Accepted Manuscript

International Journal of Modern Physics A

Article Title:	A data-efficient surrogate modeling method for a cyclotron-based proton therapy beamline based on active learning
Author(s):	Yu Chen, Bin Qin, Xu Liu, Wei Wang, Yicheng Liao
DOI:	10.1142/S0217751X24501604
Received:	14 October 2024
Accepted:	11 November 2024
To be cited as:	Yu Chen *et al.*, A data-efficient surrogate modeling method for a cyclotron-based proton therapy beamline based on active learning, *International Journal of Modern Physics A*, doi: 10.1142/S0217751X24501604
Link to final version:	https://doi.org/10.1142/S0217751X24501604

World Scientific
www.worldscientific.com

International Journal of Modern Physics A
© World Scientific Publishing Company

A data-efficient surrogate modeling method for a cyclotron-based proton therapy beamline based on active learning[*][†]

Yu Chen, Bin Qin[‡] Xu Liu, Wei Wang, Yicheng Liao

State Key Laboratory of Advanced Electromagnetic Technology, School of Electrical and Electronic Engineering, Huazhong University of Science and Technology
Wuhan 430074, Hubei, P. R. China
bin.qin@hust.edu.cn

Received Day Month Year
Revised Day Month Year

High-fidelity beamline models typically involve particle-tracking and particle-matter interaction, which are intensive computationally demanding, and time-consuming. This has led researchers to adopt data-driven surrogate models as an alternative to complex physics simulations. Training a data-driven model requires many labeled data, prompting researchers to simulate diverse beamline settings and obtain corresponding labels. However, the required dataset grows increasingly large as the number of adjustable parameters increases. Therefore, this study proposes a data-efficient surrogate modeling method that employs a student-teacher framework combined with active learning (AL) query strategies to minimize labeled samples while ensuring model accuracy. The proposed method is evaluated on the energy selection system (ESS) design of the Huazhong University of Science and Technology proton therapy facility (HUST-PTF). The results show that: (i) Training the surrogate model using 684 labeled samples selected via query strategy achieves a relative error below 5% for 90% of the samples in the test set. (ii) Compared to the beamline model built by Beam Delivery Simulation (BDSIM), the computational efficiency of the surrogate model is enhanced by a factor of $\mathcal{O}(10^7)$.

Keywords: Beamline; Deep learning; Active learning; Surrogate model; Computational efficiency.

PACS numbers:

1. Introduction

Physical simulation models are essential tools for understanding beamline design and providing data for subsequent optimization of beamline settings and commissioning. The beamline model for a cyclotron-based proton therapy facility incorporates energy modulation and beam dynamics.[1] Furthermore, high-fidelity beamline

[*] This work was supported by the National Natural Science Foundation of China (11975107) and the National Key Research and Development Program of China (No. 2016YFC0105305).
[†] Code available: https://github.com/cy1034429432/A-data-efficient-surrogate-modeling-method-for-a-cyclotron-based-proton-therapy-beamline
[‡] Corresponding author.

models should consider various realistic factors, including the geometry of the vacuum chamber, actual magnetic field distribution, and particle-matter interaction. Such models are typically built by Monte Carlo codes like Beam Delivery Simulation (BDSIM),[2] Object-Oriented Parallel Accelerator Library (OPAL),[1] etc. While these codes simulate millions of particles to accurately determine beam properties at the terminal beamline, their processing time makes direct optimization using these physics models impractical for beamline designers.

Lightweight surrogate models, designed to accelerate beam property prediction at specific beamline positions,[3,4] have been proposed to mitigate the computational burden and improve efficiency. While balancing accuracy and computational speed remains a challenge, data-driven models offer a compelling solution, typically achieving parallel computation within 10 seconds,[3,5] albeit at the cost of some precision. Such fast-predicting surrogate models provide real-time data to guide beam commissioning, particularly for challenging-to-measure beam properties like divergence, thereby mitigating limitations in current diagnostic systems. In free electron laser (FEL) systems, where designers aim to achieve user-specified beam properties,[4] long-term beam commissioning is impractical.[6] Surrogate models, coupled with optimization algorithms, offer a rapid solution for determining optimal beamline settings to achieve desired beam properties.[3]

Researchers have increasingly explored machine learning (ML) and deep learning (DL) techniques to construct surrogate models. These data-driven models are trained using extensive simulation-generated datasets. Examples include the use of a neural network (NN) to predict beam properties at the Argonne Wakefield Accelerator (AWA) linac,[3] a surrogate model combined with reinforcement learning (RL) for rapid water system control at Fermilab Accelerator Science and Technology (FAST),[4] and encoder-decoder NNs employed in the medium energy beam transport (MEBT) section of the China Accelerator Facility for Superheavy Elements (CAFe II).[7] However, training these data-driven models requires a substantial amount of labeled data, generated through time-consuming physical simulations. While these methods offer potential for speed enhancements, they do not reduce the overall number of simulations required, only shifting computational costs to an earlier stage.

Furthermore, researchers have explored strategies to optimize the efficiency of physics simulation codes or implement hardware acceleration to expedite simulations. For example, the Georges Python package utilizes the Fermi-Eyges technique to simplify particle-matter interactions during energy modulation, leading to substantial reductions in computing time.[8]

Active learning (AL) leverages diversity or uncertainty to select informative unlabeled data, aiming to optimize model performance while minimizing the number of labeled samples.[9,10] Applications of AL include a teacher-student framework integrated with query strategies for engineering designs (e.g., finite element analysis, computational fluid dynamics, propeller design),[11] Bayesian uncertainty sampling for parameter space reduction in accelerators,[12] and bidirectional AL for reducing training samples in power system stability assessment.[13] These examples demonstrate AL's potential to reduce simulation executions and enhance efficiency.

Inspired by the mentioned works,[11–13] this study proposes a data-efficient surrogate modeling method to reduce the number of required training samples. The main contributions are as follows:

(i) Building upon prior efforts on the start-to-end beamline model,[14] we develop a real-time surrogate model to predict beam properties, integrating beam dynamics and particle-matter interactions.

(ii) We develop a student-teacher framework[11] combined with specially designed AL query strategies for reducing required labeled samples (i.e. the number of simulation executions).

The remainder of this study is organized as follows: Section. 2 introduces the model setup and methodology; Section. 3 presents the results of the proposed method's verification; Section. 4 offers discussions and limitations; and Section. 5 provides conclusions.

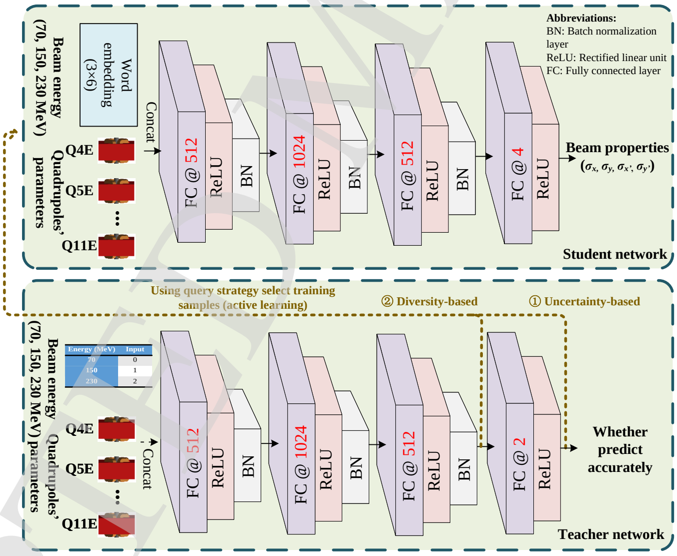


Fig. 1: Overview of the student-teacher framework and structure of the backbone model. The "@" is followed by the output size. The output of the second-to-last layer of the teacher model is used for diversity-based query strategies, while the output of the last layer is used for uncertainty-based query strategies. Word embedding, a classic deep learning module, transforms energy settings into feature representations, which are then concatenated with quadrupole parameters as input for the student network.

2. Model setup and methodology

The DL model effectively learns to predict beam properties from the simulation-generated dataset.[3,5] Fig. 1 presents an overview of the student-teacher framework and the structure of the backbone model, where the student network predicts beam properties while the teacher network evaluates whether the trained student network predicts accurately. Prior to each training iteration, the teacher network, in conjunction with AL query strategies, selects unlabeled data for labeling by the simulator.

2.1. *Selected beamline and model setup*

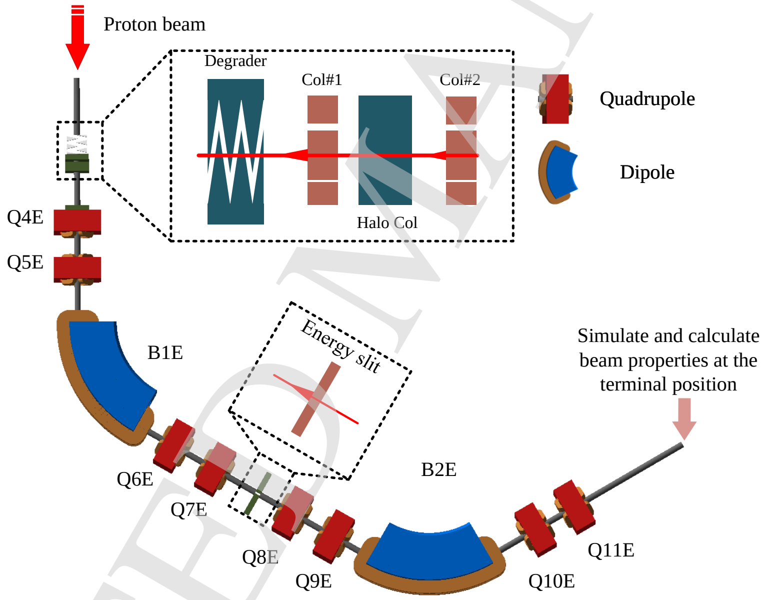


Fig. 2: Layout of ESS modeled by BDSIM.

Huazhong University of Science and Technology is currently constructing a proton therapy facility (HUST-PTF), consisting of a 240 MeV superconducting cyclotron accelerator, a transport beamline, and three treatment rooms (one fixed and two 360° rotating beams), and more detailed information can be referred to previous works.[15,16] This study selects the existing energy selection system (ESS) design for validation, consisting of 8 quadrupoles and 2 dipoles, as shown in Fig. 2. The ESS comprises a degrader with one pair of 2.5 graphite wedges for energy modulation. Following the degrader, multiple collimators and a double-bend achromatic (DBA) combined with an energy slit are employed to control the beam emittance and momentum spread. Previous work[14] has demonstrated that seamless modeling of the HUST-PTF beamline design using BDSIM yields more accurate simulation results than separate modeling, considering energy modulation and beam dynamics. Therefore, a BDSIM-based model is utilized as a simulator (i.e., data annotator) for calculating beam properties at the terminal beamline in Fig. 2, with 2,000,000 particles simulated in each run.

It should be noted that this simulator utilizes the mentioned 8 quadrupoles' (i.e., Q4E-Q11E) strengths with ± 5% range of their initial design values as parameter space and three thicknesses of the graphite wedge, corresponding to 70, 150, and 230 MeV,[17] as beam energy input, with calculated beam properties (i,e., $\sigma_x, \sigma_{x'}, \sigma_y, \sigma_{y'}$) as the label. Additionally, Sobol, a widely-used method for efficient sampling in parameter space within Bayesian statistics,[14] is used to generate 27,000 unlabeled samples within the parameter space. Among these, 5,400 samples are labeled to form a test set for evaluating the trained model's performance, while the remaining 21,600 unlabeled samples constitute an unlabeled pool for AL.

2.2. *Basic principle of AL query strategy*

Data-driven models have proven effective in constructing surrogate models for accelerators and FELs.[3–5] However, the success of these models heavily depends on the availability of sufficient high-quality labeled data. When constructing surrogate models for beamlines through simulation, designers typically generate training samples by randomly setting adjustable parameters and obtaining corresponding labels. This approach requires a significant number of samples, making the annotation process with a simulator time-consuming and impractical for labeling all unlabeled data within the parameter space. Therefore, a cost-effective method for selecting valuable samples from the unlabeled pool for annotation is crucial.

The objective of AL is to enhance model performance by minimizing the number of labeled samples required, thereby reducing data annotation costs and training time for surrogate models.[9] Fig. 3 depicts the flowchart of the pool-based AL cycle, the most common AL paradigm, with its specific procedure outlined in Algorithm 1. This study defines the unlabeled dataset $U_n^t = \{\mathcal{X}, \mathcal{Y}\}$ with n samples and the current labeled training set $L_m^t = \{X, Y\}$ with m samples. $\mathcal{X}$ and X represent the sample space of the unlabeled dataset and current labeled training set, respectively, while $\mathcal{Y}$ and Y denote the label space of the unlabeled data and current labeled training set, respectively. Additionally, $\mathbb{P}_{X,Y}$ refers to a potential distribution where $x \in \mathcal{X}, y \in \mathcal{Y}$, $\mathrm{x} \in X$, $\mathrm{y} \in Y$. The variable t denotes the number of iterations, with b samples labeled in each iteration. This study aims to design an effective AL query strategy Q and utilize the learning algorithm $\mathcal{T}$ to train a model $\mathcal{M}_{\theta_t}$, where $\mathcal{M}_{\theta_t} : X \to Y$, using as few samples or iterations (t) as possible. The optimization problem of AL can be formulated as follows:

$$\underset{t,(\mathrm{x},\mathrm{y})\in L_m^t,(x,y)\in U_n^t}{\arg\min} \mathbb{E}_{\mathrm{x},\mathrm{y}\sim\mathbb{P}_{X,Y}} \left[\ell\left(\mathcal{M}_{\theta_t}(\mathrm{x}), \mathrm{y}; \mathcal{T}; Q\right)\right] \tag{1}$$

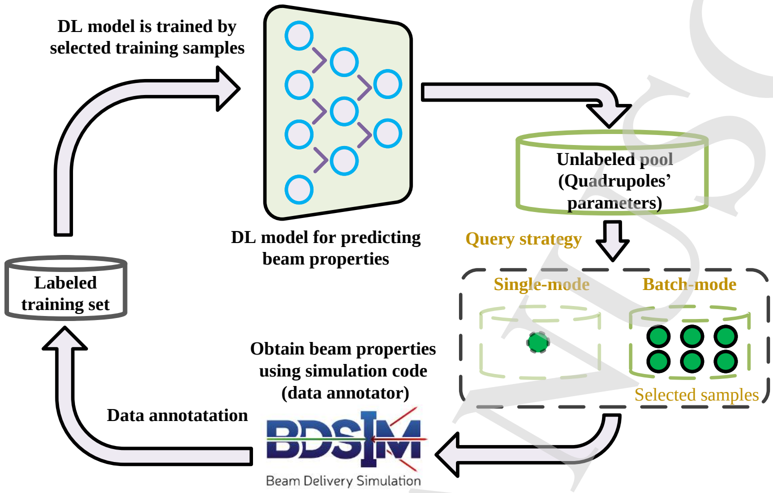

DL model is trained by selected training samples
DL model for predicting beam properties
Unlabeled pool (Quadrupoles' parameters)
Query strategy
Single-mode
Batch-mode
Selected samples
Labeled training set
Obtain beam properties using simulation code (data annotator)
Data annotatation
BDSIM
Beam Delivery Simulation

Fig. 3: General framework of pool-based AL cycle. The single-mode or batch-mode AL means using a query strategy to select one or multiple samples from the unlabeled pool for labeling.

where $\ell(\cdot)$ is the loss function.

Algorithm 1 General Pool-based AL framework.

```
Require: Unlabeled dataset $U$, current training set $L$, test set $D^T$, batchsize $b$, maximum iterations $T$, test set accuracy $Acc_{test}$, desired accuracy $Acc_{tar}$, backbone model $\mathcal{M}_\theta$, learning algorithm $\mathcal{T}$, and query strategy $Q$.
$t = 1$
Randomly select $b$ unlabeled samples and label them to construct $U^0_{n-b}$ and $L^0_b$ // Construct the initial training set. If there are existing labeled samples, they can be used as the initial training set.
$\mathcal{M}_{\theta_0} \overset{\mathcal{T}}{\leftarrow} L^0_b$ // Train $\mathcal{M}_{\theta_0}$ with initial labeled training set.
while $t \leq T$ and $Acc_{\text{test}} < Acc_{\text{tar}}$ do
    $U^t_{n-b\times(t+1)}, L^t_{b\times(t+1)} \leftarrow Q(U^{t-1}_{n-b\times t}, L^{t-1}_{b\times t}, \mathcal{M}_{\theta_{t-1}})$ // Use query strategy to select the top-$b$ unlabeled samples and hand them to simulator to label them. The current training set $L^t_{b\times(t+1)}$ consists of the old training set $L^t_{b\times t}$ and $b$ new labeled samples.
    $\mathcal{M}_{\theta_t} \overset{\mathcal{T}}{\leftarrow} L^t_{b\times(t+1)}$ // Use current labeled training set with the learning algorithm to train the data-based model.
    $Acc_{test} \overset{D^T}{\leftarrow} \mathcal{M}_{\theta_t}$ // Validating the trained data-based model on the test set.
    $t \leftarrow t + 1$
end while
return $\mathcal{M}_{\theta_t}$
```

The query strategy Q, a core component of AL, aims to select the most informative samples for training the data-driven model, thereby accelerating performance improvement. Commonly employed query strategies can be broadly categorized into three types: uncertainty-based, diversity-based, and two-stage strategies.[9, 10, 13]

Uncertainty-based query strategies select samples for labeling where the current model exhibits the highest uncertainty, aiming to provide the most informative data. These strategies identify high-uncertainty samples based on criteria such as low least confidence, high margin probability, or high entropy.

Least confidence is defined as follows:

$$x_{least} = \mathbb{P}_{\mathcal{M}_\theta}\left(\hat{y}_1 \mid x\right), x \in U \tag{2}$$

where $\hat{y}_1$ is the most probable class and $\mathbb{P}_{\mathcal{M}_\theta}\left(\hat{y}_1 \mid x\right)$ is the posterior probability prediction of $\hat{y}_1$.

Margin probability is defined as follows:

$$x_{\text{margin}} = \mathbb{P}_{\mathcal{M}_\theta}\left(\hat{y}_1 \mid x\right) - \mathbb{P}_{\mathcal{M}_\theta}\left(\hat{y}_2 \mid x\right), x \in U \tag{3}$$

where $\hat{y}_2$ is the second probable class and $\mathbb{P}_{\mathcal{M}_\theta}\left(\hat{y}_2 \mid x\right)$ is the posterior probability prediction of $\hat{y}_2$.

Entropy is defined as follows:

$$x_{\text{entropy}} = -\sum_i \mathbb{P}_{\mathcal{M}_\theta}\left(\hat{y}_i \mid x\right) \log \mathbb{P}_{\mathcal{M}_\theta}\left(\hat{y}_i \mid x\right), x \in U \tag{4}$$

where $\hat{y}_i$ ranges over all the probable classes.

Diversity-based query strategies effectively capture the distribution of the unlabeled pool by selecting representative samples. A common approach is cluster-based querying, where the unlabeled pool is divided into multiple groups based on the output of the penultimate layer, and samples are selected from each group to ensure diversity.

The k-means is a wide clustering algorithm, and the objective of k-means can be defined as follows:

$$\arg\min_{\boldsymbol{\mu}_i} \sum_{i=1, x \in U}^{k} \sum \left\|x - \boldsymbol{\mu}_i\right\|^2 \tag{5}$$

where μ_i is the cluster centers. After that, samples closest to each cluster center are selected and labeled.

Two-stage query strategies integrate uncertainty and diversity by first selecting samples based on uncertainty and then choosing diverse samples through clustering. Two common integrated query strategies are as follows:

(i) **Integrated query strategy 1**: Pre-select $r \times b$ samples based on margin probability (see Eq. (3)), where r ($r \in \mathbb{N}^+, r > 1$) is a hyperparameter to control the pre-selected sample size. From the pre-selected $r \times b$ samples, utilize k-means with $k = b$ to select b samples closest to cluster centers. The flowchart is shown in Fig. 4.

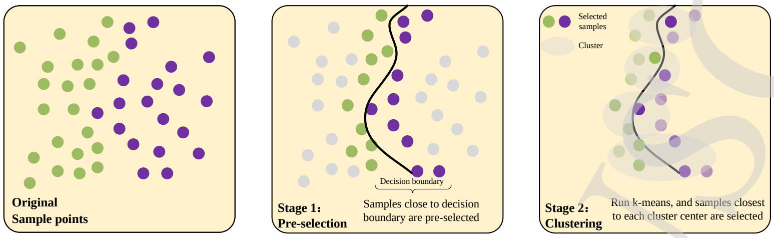

Original
Sample points
Decision boundary
Stage 1:
Pre-selection
Samples close to decision boundary are pre-selected
Selected samples
Cluster
Stage 2:
Clustering
Run k-means, and samples closest to each cluster center are selected

Fig. 4: The flowchart of two-stage query strategy.

(ii) **Integrated query strategy 2**: The entropy (4) of the unlabeled samples is calculated to serve as the sample weights of k-means with $k = b$. Subsequently, select b samples closest to the cluster centers.

2.3. *Basic principle of student-teacher framework*

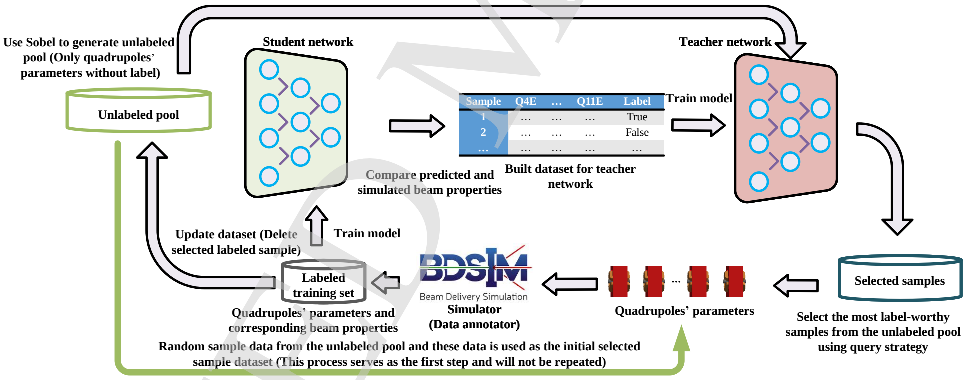

Use Sobel to generate unlabeled pool (Only quadrupoles' parameters without label)
Unlabeled pool
Student network
Teacher network
Sample
Q4E
…
Q11E
Label
1
True
2
False
Train model
Compare predicted and simulated beam properties
Built dataset for teacher network
Update dataset (Delete selected labeled sample)
Train model
Labeled training set
Quadrupoles' parameters and corresponding beam properties
BDSIM
Beam Delivery Simulation
Simulator
(Data annotator)
Quadrupoles' parameters
Selected samples
Select the most label-worthy samples from the unlabeled pool using query strategy
Random sample data from the unlabeled pool and these data is used as the initial selected sample dataset (This process serves as the first step and will not be repeated)

Fig. 5: Student-teacher framework for constructing surrogate model.

Student-teacher framework[11] employs two networks: a regressor (i.e., surrogate model) and a classifier, referred to as the student and teacher networks, respectively. The student network aims to learn the behavior of the simulator ($\mathcal{M}^{student}_{\theta_t} : X \to Y$) to predict beam properties, while the teacher network guides sampling and labeling for the next training iteration based on the performance of the current student network. During each AL iteration, both networks are trained: the student network is trained using the current labeled training set, and a custom dataset is created for the teacher network by evaluating the accuracy of the student network in predicting labeled data. To create this custom dataset, a threshold for fractional error, termed "acceptable accuracy," is defined on the test set:

$$\text{fractional error} = \frac{1}{N}\sum_{i}^{N}\frac{\left|\mathcal{Y}_i^{\text{labeled}} - \mathcal{Y}_i^{\text{predicted}}\right|}{\left|\mathcal{Y}_i^{\text{labeled}}\right|} \tag{6}$$

where N is the dimension of prediction, $\mathcal{Y}^{\text{labeled}}$ is calculated by simulator, $\mathcal{Y}^{\text{predicted}}$ is predicted by the student network.

Algorithm 2 Student-teacher framework for the surrogate model

```
Require: Unlabeled dataset $U$, current training set $L$, test set $D^T$, number of initial samples M, sample size $b$, custom training set for the teacher network $D^{labeled}$, maximum iterations $T$, test set accuracy $Acc_{test}$, desired accuracy $Acc_{tar}$, student network $\mathcal{M}_{\theta_t}^{student}$, teacher network $\mathcal{M}_{\theta_t}^{teacher}$, learning algorithm $\mathcal{T}$, and query strategy $Q$
Use the Sobel to generate $U$ in parameter space
$t = 0$
Randomly select $M$ unlabeled samples from $U$ and label them to construct $U_{n-M}^0$ and $L_M^0$
while $t \leq T$ and $Acc_{\text{test}} < Acc_{\text{tar}}$ do
    $\mathcal{M}_{\theta_t}^{student} \xleftarrow{\mathcal{T}} L_{b\times t+M}^t$ // Train the student network using the current training set and the mean square error as the loss function.
    $D_{b\times t+M}^{labeled} \xleftarrow{L_{b\times t+M}^t} \mathcal{M}_{\theta_t}^{student}$ // Create a custom training set for the teacher network.
    $\mathcal{M}_{\theta_t}^{teacher} \xleftarrow{\mathcal{T}} D_{b\times t+M}^{labeled}$ // Train the teacher network on the created custom training set using cross-entropy as the loss function.
    $Acc_{test} \xleftarrow{D^T} \mathcal{M}_{\theta_t}^{student}$ // Evaluate the student network performance on the test set.
    $t \leftarrow t+1$
    $U_{n-b\times t-M}^t, L_{b\times t+M}^t \leftarrow Q(U_{n-b\times(t-1)-M}^{t-1}, L_{b\times(t-1)+M}^{t-1}, \mathcal{M}_{t-1}^{teacher})$ // Use the teacher network combined with a query strategy to choose samples from the unlabeled pool to label.
end while
return $\mathcal{M}_{\theta_t}^{student}$
```

The user-defined threshold value represents the maximum acceptable percentage error in beam property prediction. Predictions with fractional errors below the threshold value aa are considered accurate, while those exceeding it are considered inaccurate. This study sets $aa = 0.05$ (±5% relative error) for all experiments. Based on this criterion, each sample is assigned a binary class (F), where $F = [1, 0]$ indicates an accurate prediction and $F = [0, 1]$ indicates an inaccurate prediction. Utilizing this binary class, a custom dataset $D^{labeled}$, is created for training the teacher network, which evaluates the output of the student network ($\mathcal{M}_{\theta_t}^{teacher} : X \to F$), effectively transforming the problem into a classification task. To be specific, the teacher network guides the sampling process by employing a query strategy to select samples exhibiting poor performance or a high likelihood of imprecise prediction from U, effectively targeting the limitations of the current student network. Additionally, the test set, D^T, is used to evaluate the student network's performance. The proportion of samples in D^T with relative errors less than 5% or 3% is calculated, referred to as test set Accuracy (5%) or Accuracy (3%). Throughout the learning process, the student and teacher networks are retrained using current labeled samples at each iteration until either the iteration budget is exhausted or Acc_{test} meets the specified requirement.

Table 1: Computer Hardware and Software Configuration

Device or software	Nameplate value or configuration
CPU	i9-13900K
GPU	RTX 4090
RAM	128G
Software	Python, BDSIM-v1.6.0
Package	Pytorch, Pybdsim, Tensorboard

The flowchart of the student-teacher framework is shown in Fig. 5, with the specific procedure outlined in Algorithm 2. All codes are executed on the software and hardware configurations listed in Table 1. In addition, the code and hyperparameter settings are available in the GitHub repository associated with this study.

3. Results

Using all labeled data for training is the performance upper bound (verified on the test set) of the surrogate model (i.e., student network) combined with AL. This section conducts numerous ablation experiments to explore different structures and effective training tricks for the backbone model. Subsequently, the performance of various AL query strategies is compared.

3.1. *Ablation study*

The student network is constructed using classical neural network (NN) modules, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), gated recurrent units (GRUs), and long short-term memory (LSTMs). Table 2 shows the evaluation results of these models on the test set, including mean square error (MSE), Accuracy (5%), and Accuracy (3%). As shown in Table 2, complex modules do not significantly improve the student network's performance. Conversely, employing simple fully connected layers reduces trainable parameters and yields superior performance compared to other modules. Among these models, the relatively poor performance of the CNN may be attributed to the difficulty in extracting the sequential information of quadrupoles through convolution operations. The student-teacher framework, drawing on similar concepts from the generative adversarial network (GAN),[18] benefits from maintaining parameter similarity between the student and teacher networks. Consequently, this study adopts the model structures depicted in Fig. 1.

Table 2: Student network performance between different models

Model	CNN	RNN	GRU	LSTM	**Proposed**
Trainable parameters	1068138	1149220	1149760	1150030	**1062934**
MSE	1.366	0.008	0.011	0.007	**0.006**
Accuracy (5%)	0.022	1	0.999	1	**1**
Accuracy (3%)	0.003	0.992	0.969	0.992	**0.994**

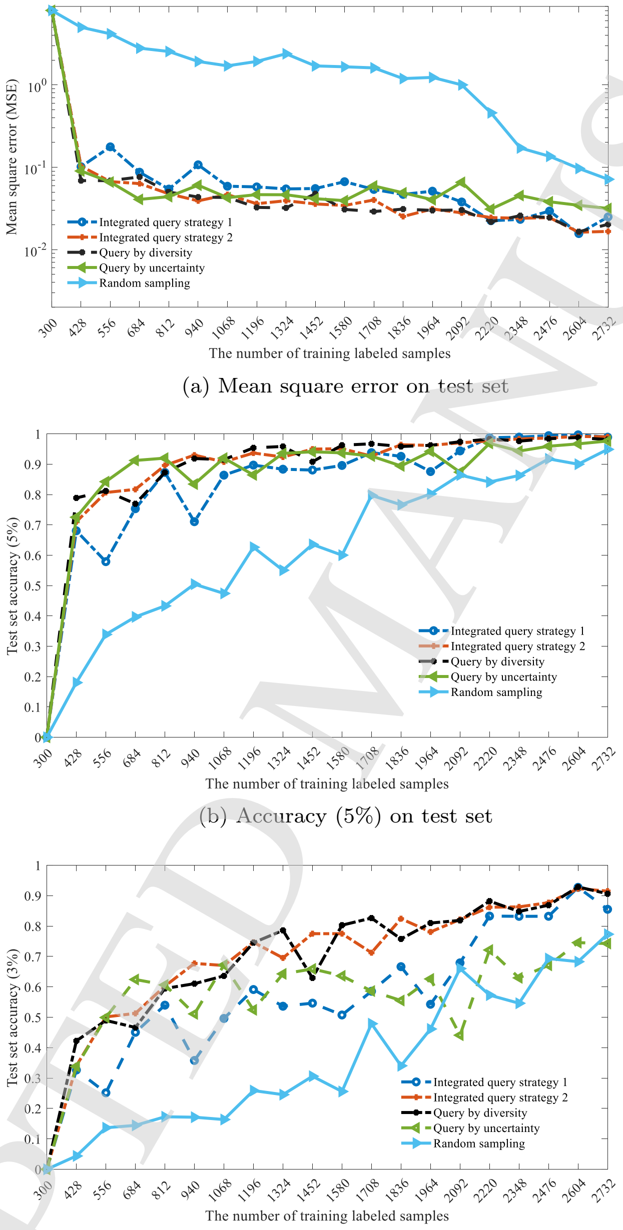
Further experiments are conducted to investigate the impact of specific training tricks on the performance of the proposed student network, as shown in Table 3. With a training epoch of 50, the initialization weight of the model significantly impacts achieving optimal results within fewer epochs. Furthermore, considering the sensitivity of beam properties to quadrupole strengths (i.e., model input), loss regularization is added to the original loss function. While word embedding and batch normalization are effective tricks, their impact on performance is less significant compared to other tricks.[18]

Table 3: Proposed student network performance without a specific trick

Tricks	MSE	Accuracy (5%)	Accuracy (3%)
With all tricks	**0.006**	**1**	**0.994**
Without batch normalization	0.008	0.999	0.976
Without word embedding	0.011	0.998	0.972
Without initialization weight	0.072	0.887	0.489
Without loss regularization	0.014	0.995	0.928

3.2. *Different query strategies*

After determining the model structures and training tricks, the student-teacher framework was implemented in conjunction with an AL query strategy to construct the surrogate model. Fig. 6 illustrates the performance of the surrogate model under various query strategies. Notably, the teacher network is a binary classification, rendering the results of Eqs. (2)-(4) nearly identical. Therefore, the least confidence metric is used to represent the uncertainty-based query strategy. Fig. 6 reveals:



(a) Mean square error on test set

(b) Accuracy (5%) on test set

(c) Accuracy (3%) on test set

Fig. 6: The evaluation of the surrogate model on the test set with specific labeled data under different query strategies.

(1) Random sampling requires a significantly larger number of labeled samples to achieve comparable performance. (2) Employing the uncertainty-based query strat-

egy to select 684 samples for training ensures that over 90% of the test set samples display a relative error of less than 5%. (3) As shown in Fig. 6c, while the uncertainty-based query strategy yields satisfactory results with minimal labeled samples, it limits further precision improvement. Therefore, an integrated query strategy (Strategy 2), considering both diversity and uncertainty, is recommended for achieving further precision gains when more labeled samples are available.

In addition, Table 4 shows three specific examples to demonstrate the predictive performance of the surrogate model trained by 684 labeled data selected by the uncertainty-based query strategy. As shown in Table 4, the relative error between the true labels and the surrogate model's outputs is below 5%.

Table 4: Comparison between the output of the surrogate model and true label

	True				Predicted			
Beam property	σ_x (mm)	σ_y (mm)	$\sigma_{x'}$ (mrad)	$\sigma_{y'}$ (mrad)	σ_x (mm)	σ_y (mm)	$\sigma_{x'}$ (mrad)	$\sigma_{y'}$ (mrad)
1	3.680	2.872	5.182	2.375	3.662	2.893	5.140	2.362
2	1.399	1.867	3.378	2.787	1.471	1.904	3.383	2.750
3	2.726	2.458	5.187	2.842	2.653	2.456	5.119	2.819

4. Discussion and limitation

Current research primarily focuses on utilizing surrogate models to accelerate beamline optimization, rather than investigating the construction of these models themselves.[3] During training data preparation, a specific range of adjustable parameters is uniformly sampled,[5] and multiple parameters are combined and labeled using a simulator. For instance, with 5 quadrupoles in the beamline, each sampled 100 times, a total of 100^5 unlabeled samples require labeling, leading to high data annotation costs. Consequently, existing approaches[3,5] do not effectively reduce computational consumption but merely shift it to the data preparation phase. In contrast to prior works, this study aims to explore the use of fewer labeled data for training a surrogate model, thereby reducing costs associated with data annotation. Furthermore, utilizing trained data-driven models can significantly accelerate the prediction of beam properties. As shown in Table 5, the surrogate model with parallel computing exhibits a speed improvement of up to $\mathcal{O}(10^7)$ compared to the BDSIM-based model (simulating 2,000k particles).

The surrogate model is constructed using the initial design strength of quadrupoles as the parameter space, defined within a relative range of 5%. An effort has also been made to construct surrogate models for a broader range of quadrupole strengths, with the results presented in Table 6. The observed poor performance is attributed to the simulator's consideration of the vacuum chamber geometry. Irrational modification of quadrupole strengths within a wider range can lead to beam loss due to particle collisions with the vacuum chamber, resulting in inaccurate

Table 5: Comparison of computing time between BDSIM-based and surrogate models. For the surrogate model, 1000 sets of experiments are conducted simultaneously to calculate the running time, demonstrating its performance of parallel computation. The BDSIM-based model is evaluated with 10 experiments to determine its running time.

The number of simulator executions	BDSIM-based model	Surrogate model
10	712min	\
1000	\	0.69s

calculation of beam properties at the terminal beamline. As depicted in Table 6, neglecting particle-matter interaction and employing optics code MAD-X[19] as the simulator for data annotation eliminates this issue. MAD-X provides deterministic results, ensuring precise calculation of beam properties at the terminal beamline without any uncertainty. It is worth noting that the previous work[14] successfully fine-tuned quadrupole strengths within a small range to improve transmission efficiency,[14] making this parameter space acceptable.

Table 6: Comparison between different parameter spaces and different simulators

	BDSIM		MAD-X	
Range of quadrupole strengths	±10%	±15%	±10%	±15%
MSE	0.084	0.094	0.026	0.032
Accuracy (5%)	0.820	0.785	0.984	0.971
Accuracy (3%)	0.500	0.491	0.943	0.822

While the proposed method demonstrates advantages in reducing required labeled data, it still has several limitations:

(i) This study develops a surrogate model for specific energy points, given the discrete thickness of the graphite block calculated from the previous work.[17] Although interpolation or fitting could be employed to determine the corresponding thickness for each energy point, errors in this process are not negligible. Therefore, the current surrogate model is limited to three characteristic energy points.

(ii) The current surrogate model applies solely to the existing ESS design at HUST-PTF, predicting beam properties only as depicted in Fig. 2. Adapting it to different beamlines necessitates the preparation of new training data. Nonetheless, the proposed methodology for constructing surrogate models is versatile and can be extended to other segments of the beamline and additional beam properties.

(iii) While BDSIM-based results are highly accurate,[20,21] this study relies solely on simulations. The extent to which these simulations accurately represent real-world measurements remains to be explored.

In future developments, referencing the latest work,[22] a digital twin of the HUST-PTF is expected to be built, interacting with the beam control system, as depicted in Fig. 7. Initially, actual measurement data from the beam diagnostic system, a subsystem of the beam control system, is used to align the digital machine with the real machine. High-fidelity physics and data-driven surrogate models can integrate additional real-world factors and be trained efficiently with limited data using transfer learning and lifelong learning, respectively, to further enhance their alignment with the real machine. Furthermore, optimization algorithms can then adjust model parameters to achieve desired beam properties. Once the digital twin is constructed, predictions or virtual measurements from the digital twin can guide user interventions in controlling the real machine, facilitating orbit correction and optics tuning. Additionally, advanced control technologies, such as reinforcement learning (RL) or advanced control theory, can be employed to enable interaction between the real machine and the beam control system.

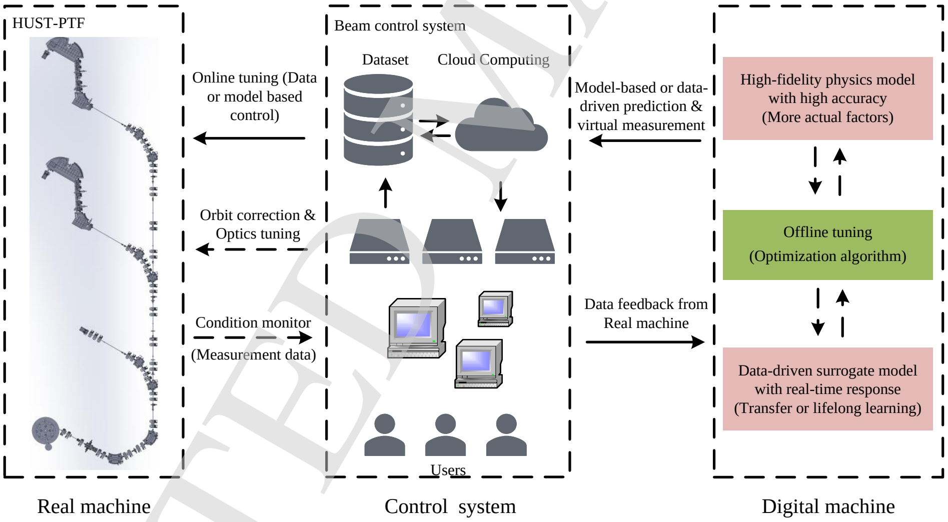

HUST-PTF
Online tuning (Data or model based control)
Orbit correction & Optics tuning
Condition monitor
(Measurement data)
Beam control system
Dataset
Cloud Computing
Users
Model-based or data-driven prediction & virtual measurement
Data feedback from Real machine
High-fidelity physics model with high accuracy (More actual factors)
Offline tuning (Optimization algorithm)
Data-driven surrogate model with real-time response (Transfer or lifelong learning)
Real machine
Control system
Digital machine

Fig. 7: Future prospects of the digital twin for the HUST-PTF

5. Conclusion

This study proposes a data-efficient surrogate modeling method for a cyclotron-based proton therapy beamline. Based on the above results, the following conclusions are drawn:

(i) Adopting AL can significantly reduce the number of simulator executions. While the uncertainty-based query strategy achieves promising results with fewer labeled data, it limits the potential for higher accuracy. Incorporating a two-stage query strategy that considers both uncertainty and diversity proves beneficial for constructing a more accurate surrogate model.

(ii) Replacing the physics-based beamline simulation with a data-driven surrogate model can significantly enhance the speed and computational efficiency of obtaining beam properties.

Acknowledgement

We thank Dr. Zhongtuo Shi at the School of Electrical and Electronic Engineering, Huazhong University of Science and Technology (HUST), who gave helpful suggestions about applying AL.

References

1. V. Rizzoglio, A. Adelmann, C. Baumgarten, M. Frey, A. Gerbershagen, D. Meer and J. M. Schippers, *Phys. Rev. Accel. Beams* **20**, 124702 (Dec 2017).
2. L. Nevay, S. Boogert, J. Snuverink, A. Abramov, L. Deacon, H. Garcia-Morales, H. Lefebvre, S. Gibson, R. Kwee-Hinzmann, W. Shields and S. Walker, *Comput. Phys. Commun.* **252**, 107200 (2020).
3. A. Edelen, N. Neveu, M. Frey, Y. Huber, C. Mayes and A. Adelmann, *Phys. Rev. Accel. Beams* **23**, 044601 (Apr 2020).
4. A. L. Edelen, S. G. Biedron, B. E. Chase, D. Edstrom, S. V. Milton and P. Stabile, *IEEE Trans. Nucl. Sci.* **63**, 878 (2016).
5. A. Edelen, S. G. Biedron, J. P. Edelen, S. V. Milton and P. J. van der Slot, Using a neural network control policy for rapid switching between beam parameters in an fel (2017).
6. X. Yang, Y. Chen, J. Wang, H. Zheng, H. Liu, D. Zhou, Y. He, Z. Wang and Q. Zhou, *Ann. Nucl. Energy* **179**, 109346 (2022).
7. K. Sun, X. Chen, X. Zhao, X. Qi, Z. Wang and Y. He, *Int. J. Mod. Phys. A* **38**, 2350145 (2023).
8. R. Tesse, C. Hernalsteens, E. Gnacadja, N. Pauly, E. Ramoisiaux and M. Vanwelde, *SoftwareX* **24**, 101579 (2023).
9. B. Settles, *Active Learning Literature Survey*, Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009).
10. P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen and X. Wang, *ACM Comput. Surv.* **54** (Oct 2021).
11. H. Vardhan, U. Timalsina, P. Volgyesi and J. Sztipanovits, Data efficient surrogate modeling for engineering design: Ensemble-free batch mode deep active learning for regression (2022).
12. R. Roussel, J. P. Gonzalez-Aguilera, Y.-K. Kim, E. E. Wisniewski, W. Liu, P. Piot, J. Power, A. Hanuka and A. Edelen, *Nat. Commun.* **12** (2021).
13. Z. Shi, W. Yao, Y. Tang, X. Ai, J. Wen and S. Cheng, *IEEE Trans. Power Syst.* **38**, 5128 (2023).
14. Y. Chen, B. Qin, X. Liu, W. Wang and Y. Liao, *Nucl. Eng. Technol.* **56**, 4365 (2024).
15. B. Qin, X. Liu, Q.-S. Chen, D. Li, W.-J. Han, P. Tan, Z.-Q. Zhang, C. Zhou, A.-T. Chen, Y.-C. Liao and W. Wang, *Nucl. Sci. Tech.* **32** (12 2021).
16. P. Li, D. Li, B. Qin, C. Zhou, W. Han, Y. Liao and A. Chen, *Nucl. Eng. Technol.* **54**, 2852 (2022).

17. Z. Liang, K. Liu, B. Qin, W. Chen, X. Liu, D. Li and Y. Xiong, *Nucl. Instrum. Methods Phys. Res., Sect. A* **890**, 112 (2018).
18. Y. Chen, Z. Zhao, J. Liu, S. Tan and C. Liu, *Eng. Fail. Anal.* **159**, 108115 (2024).
19. `https://madx.web.cern.ch/`,.
20. V. Maradia, A. C. Giovannelli, D. Meer, D. C. Weber, A. J. Lomax, J. M. Schippers and S. Psoroulas, *Med. Phys.* **49**, 2183 (2022).
21. V. Maradia, D. Meer, D. C. Weber, A. J. Lomax, J. M. Schippers and S. Psoroulas, *Med. Phys.* **48**, 7613 (2021).
22. A. Edelen, C. Mayes, C. Emma and et al., Progress on combining digital twins and machine learning-based control for accelerators at slac, in *Proc. 15th International Particle Accelerator Conference*, (05 2024), pp. 1843–1846.