Large-Scale Optimizations in Proton Beam Radiotherapy by Neural Network Denoising of Robust Simulated Patient Data

Angelo Calingo*
Department of Computer Science & Engineering
University of Nevada, Reno, USA

Bikash Gautam*
Department of Computer Science
Alabama Agricultural and Mechanical University, USA

Peter L. Jin*
James M. Bennett High School
Salisbury, MD, USA

Sidhya Pathak*
Department of Computer Science
University of Virginia, USA

Michelle Zhao*
Department of Computer Science
Cornell University, USA

Hussam Fateen
Dept. of Mathematics and Statistics
U. of Maryland, Baltimore County, USA

Harrison Lewis
Dept. of Mathematics and Statistics
U. of Maryland, Baltimore County, USA

Matthias K. Gobbert
Dept. of Mathematics and Statistics
U. of Maryland, Baltimore County, USA

Vijay R. Sharma
Department of Radiation Oncology
U. of Maryland School of Medicine, USA

Lei Ren
Department of Radiation Oncology
U. of Maryland School of Medicine, USA

Ananta Chalise
Department of Radiation Oncology
U. of Maryland School of Medicine, USA

Stephen W. Peterson
Department of Physics
U. of Cape Town, South Africa

Jerimy C. Polf
M3D, Inc.
USA

***Abstract*—Proton beam radiotherapy is an advanced cancer treatment utilizing high-energy protons to destroy tumor matter. This treatment requires precise Bragg-peak localization, but Compton-camera image reconstructions are often unusable due to mischaracterized scattering sequences and excessive image noise. We present machine learning models to classify the scattering events. Multiple novel robust-volume datasets simulating particle interactions with human tissue were generated using Duke University CT scans and Geant4 and Monte-Carlo Detector Effects (MCDE) software. Novel implementations of an Event Classifier Transformer and a 1D Convolutional Neural Network (CNN) were developed, while prior models, such as Fully-Connected Neural Networks (FCN) and Long Short-Term Memory Neural Network (LSTM), were optimized through large-scale hyperparameter studies using a novel automated tuning framework built into the Big-Data REU Integrated Development and Experimentation (BRIDE) machine learning pipeline. Fully-connected neural networks and convolutional neural networks show significant improvements in model accuracy over prior work on simulated patient data and demonstrate that relatively shallow, regularized models generalize best.**

***Index Terms*—Proton beam therapy, Compton camera, Classification, Recurrent neural network, PyTorch**

I. Introduction

Proton beam radiotherapy is an advanced cancer treatment that utilizes high-energy protons to destroy tumor matter. In the therapy, protons deposit the vast majority of their energy in a localized area called the Bragg peak. This Bragg peak property significantly reduces unnecessary radiation exposure to healthy tissue surrounding the tumor compared to other radiation therapy techniques such as X-ray radiotherapy [8], [11]. In a clinical setting, medical professionals need to accurately determine the location of the proton beam and thus the Bragg peak to ensure that healthy tissue is not damaged by radiation.

When the proton beam interacts with patient medium, prompt gamma rays are emitted and can be detected by an imaging device called the Compton camera. Through the Compton scattering process, the camera can determine the gamma ray path and thus the Bragg peak location. However, due to the non-zero time resolution of the Compton camera, the prompt gamma interactions may be detected in an incorrect order, leading to severe image noise in the subsequent image reconstruction of the beam path and rendering the images unusable for determining the location of the Bragg peak with respect to the patient's body. Machine learning is utilized to correctly sequence the order of the scattering events to remove noise from reconstructed images, in order to accurately deduce the true location of the Bragg peak. Deep learning methods including feedforward models, recurrent models, and transformers are specifically developed for this purpose.

Simulated data sets must be utilized in the neural network

*These authors contributed equally to this work.

2375-9259/25/$31.00 ©2025 IEEE
DOI 10.1109/ICDMW69685.2025.00356

development process due to the ethical constraints of capturing scatterings with actual patients in a laboratory. Previous research based data on variable-density patient CT scan measurements but has been plagued by low model accuracy results due to small dataset sizes [6].

To improve the denoising of scattering data for Compton Camera image reconstruction in proton beam therapy, this work makes the following contributions:

- Generating multiple significantly larger novel simulated patient datasets for model training of up to eight times those used in previous work.
- Developing novel implementations of spatially-suited neural networks including the Event Classifier Transformer and 1D Convolutional Neural Network, and testing other models such as the Fully-Connected Neural Network and Long Short-Term Memory Neural Network.
- Efficiently boosting model robustness through performing multiple automatic, large-scale grid searches for hyperparameter tuning.

The code of this work is publicly available in a GitHub repository [3].

This paper is structured as follows: Section II provides background information on proton beam radiation therapy, scatter imaging, and the motivation for applying machine learning. Section III details the specific machine learning techniques used within this study, related works, the data generation process, and the modular coding platform utilized for hyperparameter tuning. Section IV presents the models' performance results on the novel simulated patient data under grid search hyperparameter optimization. Section V explores additional studies conducted on another novel simulated patient dataset. Section VI summarizes key findings of our work. Finally, Section VII discusses potential future directions that may be promising.

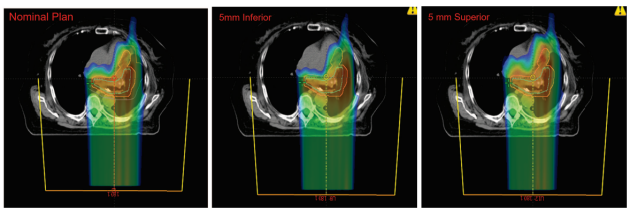
II. Application Background

A. Proton Beam Radiotherapy

Proton beam therapy is a type of external beam radiation therapy that uses high-energy proton beams to deposit radiation at cancer tumor sites, destroying cancerous tissue. Unlike many other radiation therapies which deliver unnecessary radiation at healthy tissue sites near the tumor, proton beam therapy deposits the majority of the radiation at a localized peak called the Bragg peak. During treatment, the Bragg peak must be located directly on the tumor to deposit radiation precisely at the correct location to avoid damaging healthy tissue; it cannot undershoot or overshoot the tumor, as seen in Fig. 1. However, to do this, the location of the Bragg peak needs to be determined exactly.

B. Compton Camera Image Reconstruction

When protons in the beam interact with patient matter, prompt gamma rays are emitted and can be captured by the Compton camera imaging device [13]. When these prompt gammas interact with the camera, the device captures the scattering's spatial (x_i, y_i, z_i) coordinates and energy level



(a) Normal (b) Undershoot (c) Overshoot

Fig. 1: Range and location uncertainties of the proton beam and Bragg peak. (a) shows the Bragg peak exactly on the tumor, (b) shows the beam undershooting the tumor, and (c) shows the beam overshooting the tumor [10].

(e_i). From this captured scattering data, the Compton cone of emission with multiple detectors inside the camera can be used to statistically reconstruct the location of the source of the prompt gammas [12]. The prompt gamma origin locations can then be used to visually determine the path of the proton beam and the Bragg peak location through image reconstruction algorithms. However, because of the finite time resolution of the Compton camera, gamma scatterings are detected almost simultaneously and thus in the incorrect sequential order, leading to severe noise in the reconstructed images and rendering them blurry and unusable for determining whether the Bragg peak is on the tumor location.

1) Scattering Types: There are multiple types of scattering events detected by the Compton camera; in order to separate out false events that create image noise due to mischaracterized scattering sequences, the following groupings are used:

1) True Triples: these consist of three sequential interactions with the Compton camera and occur in six distinct orderings (123, 132, 213, 231, 312, 321). Only the 123 ordering can be used for image reconstruction.
2) Doubles to Triples (DtoT): these consist of a double interaction and an independent single interaction that are detected as the same scattering. There are six distinct orderings of DtoTs (124, 134, 214, 234, 324, 314). The "4" signifies to the second prompt gamma interaction during misdetection.
3) False Triples: these consist of three independent events that are falsely detected as a single scattering. They result in noisy images and must be discarded during image reconstruction.

C. The Need for Machine Learning

Traditional methods have attempted to improve image reconstruction quality by using filters. In [9], images were cleaned by obtaining threshold values calculated from the distance from the center of an image pixel to the Compton cone. However, filters have limitations including having a high probability of ignoring complex patterns of noise or keeping false events. Machine learning, in contrast, capable of identifying much more complex patterns than a filter can. Machine learning models are capable of accurately classifying the various scatter events from the Compton camera and

removing false events from the data, leading to a de-noised image.

III. Methodology

A. Machine Learning

Our approach focuses on using machine learning to classify prompt gamma scattering events detected by a Compton camera during proton beam therapy. The goal is to improve real-time treatment verification with image reconstruction by identifying the scatter type based on spatial and energy features from three detected interactions. We implemented and evaluated several neural network architectures, including fully connected models, recurrent models, convolutional models, and transformers.

1) Fully Connected Neural Networks: Fully Connected Networks (FCNs) are feedforward models in which each neuron in one layer is connected to all neurons in the next. These models treat the input as a fixed-length vector and are well suited for structured data. The FCN processes the input data through multiple dense layers to learn patterns that distinguish the classes. The final layer uses softmax activation to predict the most likely class. FCNs are relatively fast to train and perform reliably on tabular data like ours.

2) Recurrent Neural Networks: We used Long Short-Term Memory (LSTM) networks, a type Recurrent Neural Network (RNN), to explore whether modeling ordered relationships across the three interactions could improve classification. LSTM models maintain memory over longer sequences by using gated cells; an input gate determines which information should be stored in the cell, a forget gate determines what should be discarded, and an output gate controls what information should be outputted. In the model, the 15 input features are processed as a sequence grouped by interaction, and the LSTM captures how spatial or energy patterns evolve across hits. This structure allows the model to retain useful context between interactions. The LSTM output is passed to fully connected layers to make a final prediction.

3) Convolutional Neural Networks: To capture localized patterns with the feature vector, a novel implementation of a one-dimensional Convolutional Neural Network (1D CNN) was developed. This architecture applies filters across the input features to extract spatial and energetic correlations between the detected hits. The model consists of seven stacked Conv1D layers with ReLU activations; pooling layers are placed between them and are followed by fully connected layers for final classification. In our task, the CNN effectively captures spatial patterns across the three interactions and was thus thought to be suitable for distinguishing between scatter types.

4) Transformer Neural Networks: To better model global feature dependencies, we implemented a novel transformer-based architecture known as the Event Transformer Classifier. Each of the 15 input features was treated as an individual token and passed through an embedding layer. The resulting sequence was processed by multiple transformer encoder blocks containing self-attention and feedforward layers. A learnable class token was added at the beginning to aggregate contextual information and positional encoding was used to preserve feature order. This architecture helped the model learn feature interactions across all three hits, including patterns between energy and spatial values that other models may miss.

B. Related Works

In prior research, machine learning has been shown to be an appropriate tool for de-noising prompt gamma radiation image reconstruction. [5] delved into its use on both simple, constant-density water phantom and the more complex, variable-density simulated patient data. Their best performing model on water phantom data was a residual FCN with a 75% accuracy. However, for patient data, their best model (a 4-layer LSTM) achieved an accuracy of only 55.6%. They noted that the accuracy was limited likely due to a small patient dataset size, the apparent complexities of patient data, and a restricted breadth of hyperparameter studies.

All other studies besides [5] have been conducted on water phantom data. [1] explored multiple neural network architectures and achieved a greatest validation accuracy of 75% using an FCN. The authors of [7] primarily explored recurrent models with a highest accuracy of 73% but had significantly simplified model architectures with faster loading and prediction times for a clinical scenario. Finally, the research in [2] transitioned models into a flexible Pytorch architecture compared to the previously utilized Tensorflow, with slightly lower peak accuracies of 69%.

C. Dataset Generation

The data generation process began by using the Geant4 C++ toolkit to simulate the interactions of protons with patient matter based on Duke University CT measurements. Geant4 produced raw event data including spatial coordinates and energy levels of each gamma interaction. This data was then passed through Monte Carlo Detector Effects (MCDE) software to add detector timing and trigger effects to make the generated data more realistic to actual Compton camera captured data. MCDE produced scattering type information and class labels; after filtering for only triples scatterings, the data was normalized and shuffled. Final datasets consist of 15 feature columns with 3 sets of energy and spatial coordinates (e_i, x_i, y_i, z_i) and 1 set of Euclidean distances between coordinates (euc_1, euc_2, euc_3).

The 1.1 million observation dataset was created by processing each Geant4 energy layer data separately through MCDE at six different proton dosage rates (1, 20, 40, 140, and 180 kMU/min). In contrast, the 4.1 million observation dataset involved combining all energy layers before running MCDE at a fixed dose of 100 kMU/min; this led to a much larger dataset.

Compared to constant-density water phantom and variable-density patient datasets from previous work of which had fewer observations or less realistic setups, these two novel introduced datasets are both much larger and more similar to

clinical data because simulated patient data involves variable-density measurements that emulate actual clinical patient medium [6], [10]. The increased data size allowed us to train better models and run more thorough hyperparameter studies with higher confidence in the results. Note that additional efforts are underway to collect further realistic patient CT measurements and data.

D. The BRIDE Platform

In this work, we utilized the Big-Data REU Integrated Development and Experimentation (BRIDE) platform from [6] and enhanced it with a newly integrated hyperparameter tuning technique through the use of grid search. BRIDE is a Pytorch Lightning framework for developing machine learning models and includes data processing, model coding, job submission, and rigorous model evaluation. Its structure is shown in Fig. 2. Note that all BRIDE code is publicly available in [3] in the folder `2025-projects/team-2`.

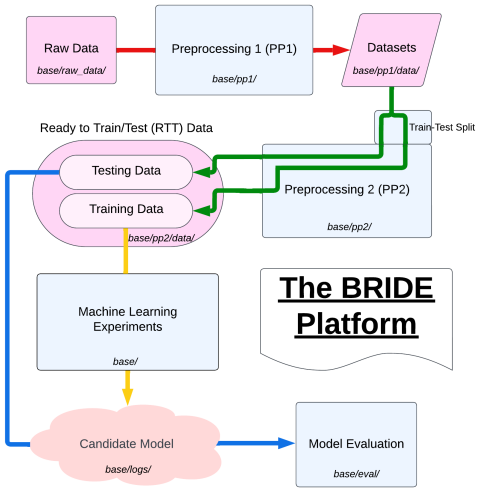


Fig. 2: Flowchart of the structure of the BRIDE platform [6].

Previously, grid search on BRIDE functioned by submitting runs separately for each different hyperparameter configuration. However, for conducting extensive model tuning, this method quickly became both inefficient and time-consuming. To address this, a grid search script was developed to automate and accelerate the hyperparameter tuning process. The developed grid search functions through a `sklearn ParameterGrid` that generates all possible combinations of hyperparameters that have been given by the user; for each combination, new run directories and YAML configuration files are created and a Slurm job for the model training process is submitted. All machine learning experiments in this work were conducted through the use of grid search. To choose the candidate model for later evaluation on the test set, only the run with the best result was needed.

E. Hardware and Software

For this work, we utilized Graphics Processing Units (GPU) in the chip cluster maintained by the UMBC High Performance Computing Facility (hpcf.umbc.edu). The GPU portion of the chip cluster consists of four nodes with eight RTX 2020Ti GPUs (11 GB of GDDR6 memory each), seven nodes with eight RTX 6000 GPUs (24 GB of GDDR6 memory each), two nodes with eight RTX 6000 GPUs (48 GB of GDDR6 memory each), two nodes with two H100 GPUs (100 GB of HBM3 memory each), and ten nodes with four L40S GPUs (48 GB of GDDR6 memory each).

The machine learning models were built and implemented using PyTorch v2.3.1 (https://PyTorch.org). For data preprocessing and manipulation, we used scikit-learn v1.3.0 (https://scikit-learn.org/stable/), pandas v2.2.2 (https://pandas.pydata.org/), and numpy v1.26.4 (https://numpy.org/). To visually display our results, we used matplotlib v3.8.4 (https://matplotlib.org/) and seaborn v0.13.2 (https://seaborn.pydata.org/). Our models were built inside of the python environment Anaconda3 (https://www.anaconda.com/).

IV. Results

A. Patient Data

To improve the generalizability of machine learning models in the data sequencing process, hyperparameter studies were conducted on on the novel 4.1 million row simulated patient data using the newly developed grid search mechanism in BRIDE.

1) FCN Hyperparameter Study: The FCN hyperparameter study tested 18 combinations of hyperparameters by varying neuron configuration, dropout, and learning rate as seen in Table I. Certain parameters were held constant throughout the grid search as displayed in Table II. Note that all runs implemented early stopping of 1500 epochs to avoid unnecessary runtime and any possible decreases in validation accuracy after plateauing. The results of this hyperparameter study are shown in Table III. Note that for this and all following hyperparameter studies, accuracy on the test set was only found for the model with the highest validation accuracy.

Hyperparameter	Value
Neuron Configuration	[512, 256, 256, 256, 256, 256, 256, 128] [256, 256, 128, 128]
Dropout	0.05, 0.2, 0.4
Learning Rate	0.001, 0.0005, 0.0001

TABLE I: FCN grid search candidate hyperparameters.

The best model from this FCN hyperparameter study achieved training, validation, and testing accuracies of 72.1%, 74.2%, and 74.2%, respectively. The training and validation curves can be found in Fig. 3, and the test set confusion matrix can be found in Fig. 4. The hyperparameters for this model include a neuron configuration of [512, 256, 256, 256, 256, 256, 256, 128], dropout of 0.05, and a learning rate of 0.0001. These accuracy results are significant as they constitute a large 18% numerical increase in testing accuracy from any

Hyperparameter	Value
Hardware	4 rtx6000 GPUs
Validation Split	0.1
Batch Size	256
Learning Rate Gamma	0.9
Learning Rate Step	100
L2	0.01
Loss Function	CrossEntropy + Custom Pairwise Loss
Optimizer	Adam
Activation Function	ReLU

TABLE II: FCN grid search constant hyperparameters.

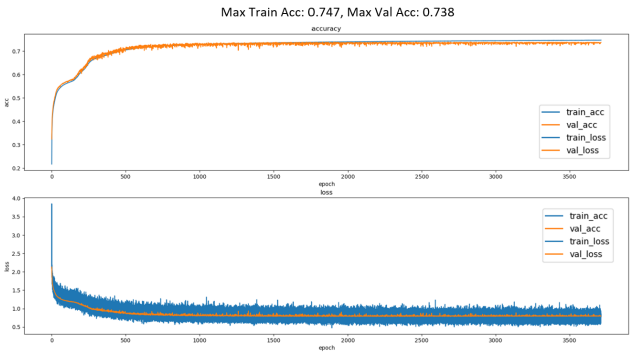


Fig. 3: Training and validation accuracy and loss curves for the best FCN model in the 4.1 million row simulated patient data hyperparameter study.

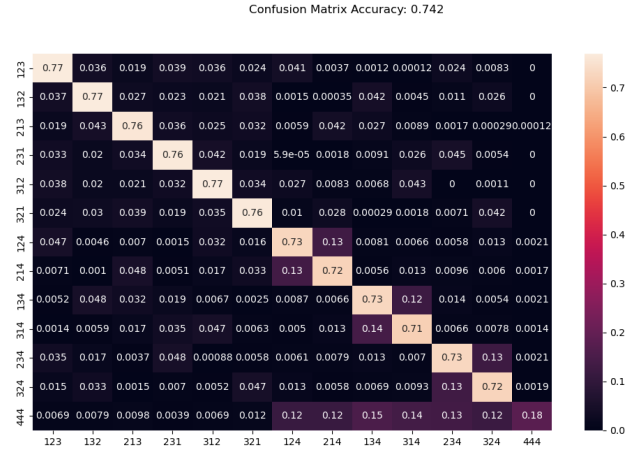


Fig. 4: Confusion matrix on the test set for the best FCN model on the 4.1 million row simulated patient hyperparameter study.

previous machine learning work on simulated patient data [6]. Additionally, this achieved testing accuracy is also 1.7% numerical increase from the best results on the much simpler water phantom data from previous BRIDE research [6].

From this hyperparameter study, we conclude that while deeper architectures tended to suffer from overfitting, proper regularization techniques led to better results than with shallower networks. However, without enough regularization,

Test	Max Train Acc.	Max Val. Acc.	Test Acc.
1	0.712	0.716	-
2	0.723	0.727	-
3	0.738	0.742	0.742
4	0.708	0.728	-
5	0.714	0.728	-
6	0.720	0.736	-
7	0.560	0.578	-
8	0.682	0.713	-
9	0.702	0.727	-
10	0.555	0.576	-
11	0.557	0.578	-
12	0.673	0.716	-
13	0.523	0.554	-
14	0.536	0.564	-
15	0.541	0.569	-
16	0.518	0.552	-
17	0.519	0.553	-
18	0.521	0.556	-

TABLE III: Results of the FCN hyperparameter study on the 4.1 million row simulated patient dataset.

deeper networks performed up to 20% worse in accuracy. Learning rate had a minimal impact on performance but a low learning rate tended to perform the best. It is clear that no overfitting is observed with this model likely due to dropout and L2 regularization, a significant improvement over previous results where severe overfitting took place [6].

2) LSTM Hyperparameter Study: The LSTM hyperparameter study tested 18 hyperparameter configurations. The neuron configurations of the fully-connected layers, dropout, and learning rate were varied, as shown in Table IV. Throughout the study, several parameters were held constant as detailed in Table V. The runs took place with early stopping set to 1500 epochs. The results of this study are listed in Table VI.

Hyperparameter	Value
Neuron Configuration	[128,128,128,128], [128,64]
Dropout	0.05, 0.15, 0.3
Learning Rate	0.001, 0.0005, 0.0001

TABLE IV: LSTM grid search candidate hyperparameters.

Hyperparameter	Value
Hardware	4 rtx6000 GPUs
Validation Split	0.1
Batch Size	256
Learning Rate Gamma	0.1
Learning Rate Step	1000
L2	0.0000001
Loss Function	CrossEntropy + Custom Pairwise Loss
Optimizer	Adam
Activation Function	ReLU
Max Epochs	4000

TABLE V: LSTM grid search constant hyperparameters.

The best-performing model achieved training, validation, and testing accuracies of 74.7%, 73.7%, and 73.7%, respectively. Aaccuracy and loss curves are displayed in Fig. 5, and the test set confusion matrix is shown in Fig. 6. Optimal hyperparameters included a 2-layer fully-connected layer architecture of [128, 64], dropout of 0.3, and learning rate of

0.0001. Significantly, the achieved LSTM testing accuracy is an approximately 18% numerical increase in accuracy from the highest result on simulated patient data from any previous work [6]. The model's robustness in terms of testing accuracy results is also comparable to the best achieved BRIDE result on water phantom from previous research [6]. In fact, it was slightly higher (1.2% numerical increase) at 73.7% compared to 72.5% even though previous work was on simpler data.

Through the study, we found that a shallower architecture of 2 fully-connected layers performed significantly better than the deeper architecture of 4 fully-connected layers. Although fully-connected layers only map the LSTM layers' output to the output space, a fewer number of those layers and neurons may have offered advantages in terms of limiting overfitting and increasing model generalizability to new data, as shallower layers reduced the model's capacity to memorize the training data.

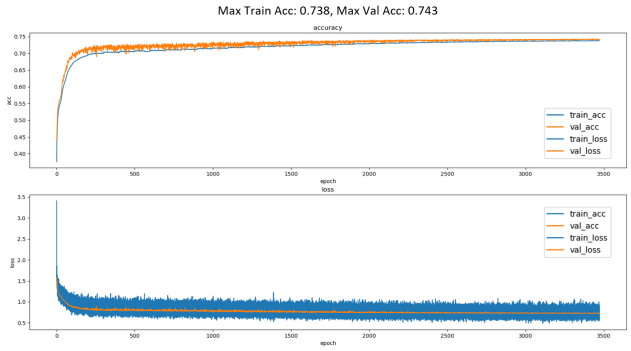


Fig. 5: Training and validation accuracy and loss curves for the best LSTM model on the 4.1 million row simulated patient hyperparameter study.

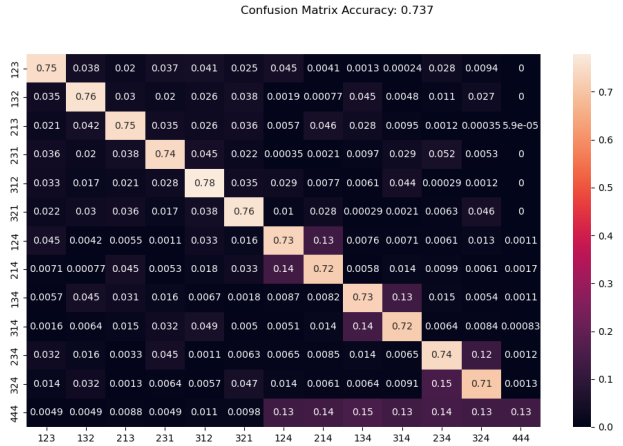


Fig. 6: Confusion matrix on the test set for the best LSTM model on the 4.1 million row simulated patient hyperparameter study.

Test	Max Train Acc.	Max Val. Acc.	Test Acc.
1	0.75	0.723	-
2	0.753	0.713	-
3	0.735	0.729	-
4	0.747	0.715	-
5	0.749	0.728	-
6	0.749	0.731	-
7	0.736	0.721	-
8	0.739	0.729	-
9	0.747	0.737	0.737
10	0.748	0.723	-
11	0.751	0.726	-
12	0.747	0.735	-
13	0.745	0.721	-
14	0.749	0.726	-
15	0.750	0.732	-
16	0.732	0.721	-
17	0.741	0.718	-
18	0.741	0.725	-

TABLE VI: Results for the LSTM hyperparameter study on the 4.1 million row simulated patient dataset.

3) Transformer Hyperparameter Study: This hyperparameter study for the Event Classifier Transformer neural network tested 32 combinations of hyperparameters by varying learning rate, dropout, number of model nodes, number of layers, and number of heads as shown by Table VII. Several hyperparameters were held constant throughout the experiments and are shown in Table VIII, and early stopping was set to 1500 epochs. The accuracy results of this study are listed in Table IX.

Hyperparameter	Value
Model Nodes	128,256
Number of Layers	4, 6
Number of Heads	4, 8
Dropout	0.15, 0.3
Learning Rate	0.0005, 0.0001

TABLE VII: Transformer grid search candidate hyperparameters.

Hyperparameter	Value
Hardware	4 rtx6000 GPUs
Validation Split	0.1
Batch Size	256
Learning Rate Gamma	0.1
Learning Rate Step	1000
L2	0.0000001
Loss Function	CrossEntropy + Custom Pairwise Loss
Optimizer	Adam
Activation Function	ReLU
Max Epochs	4000

TABLE VIII: Transformer grid search constant hyperparameters.

For this study, the highest-performing transformer model had a training, validation, and testing accuracy of 75.1%, 70.5%, and 70.4%, respectively. The training and validation curves and the confusion matrix on the test set are shown in Figs. 7 and 8, respectively. The model's hyperparameters

Test	Max Train Acc.	Max Val. Acc.	Test Acc.
1	0.751	0.705	0.704
2	0.775	0.664	-
3	0.780	0.663	-
4	0.900	0.646	-
5	0.762	0.675	-
6	0.781	0.644	-
7	0.844	0.641	-
8	0.917	0.639	-
9	0.721	0.679	-
10	0.638	0.658	-
11	0.473	0.653	-
12	0.253	0.615	-
13	0.697	0.659	-
14	0.613	0.657	-
15	0.393	0.618	-
16	0.314	0.612	-
17	0.707	0.653	-
18	0.717	0.614	-
19	0.731	0.658	-
20	0.752	0.601	-
21	0.713	0.614	-
22	0.721	0.568	-
23	0.741	0.625	-
24	0.782	0.638	-
25	0.713	0.617	-
26	0.743	0.593	-
27	0.769	0.610	-
28	0.814	0.579	-
29	0.736	0.585	-
30	0.747	0.609	-
31	0.781	0.599	-
32	0.828	0.592	-

TABLE IX: Results of the transformer hyperparameter study on the 4.1 million row simulated patient dataset.

included 128 model nodes, 4 layers, 4 model heads, dropout of 0.15, and a learning rate of 0.0005. The optimal transformer's testing accuracy is importantly a 14% numerical increase from the best result on simulated patient data in other work [6]. We found that the transformer model accuracy was sometimes extremely variable, and the model also experienced severe overfitting with up to a 30% difference between training and validation accuracy. However, dropout did assist in addressing overfitting. Manipulating the patience to be lower could aid in preventing unnecessary runtime after the model accuracy plateaued.

V. Additional Studies

In addition to hyperparameter studies on the 4.1 million row simulated patient dataset, similar studies on the 1.1. million row simulated patient dataset were also conducted for the LSTM, FCN, and 1D CNN models. For each model, 18-24 hyperparameter configurations were tested, with neuron configuration, learning rate, dropout, and batch size being varied. Table X summarizes the results from these studies. More detailed model results can be found in [4].

In the study, the FCN model performed slightly better than the LSTM in terms of validation and testing accuracy, which was also the case for the grid search on the 4.1 million row patient dataset. This suggests that FCNs may be more suited for learning complex patterns from data based on variable-density patient medium. Both the FCN and LSTM performed

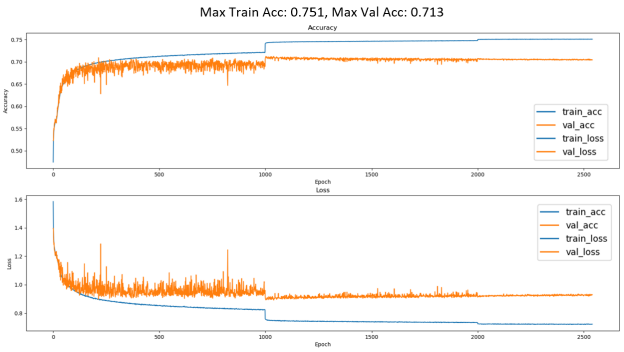


Fig. 7: Training and validation accuracy and loss curves for the best transformer model on the 4.1 million row simulated patient hyperparameter study.

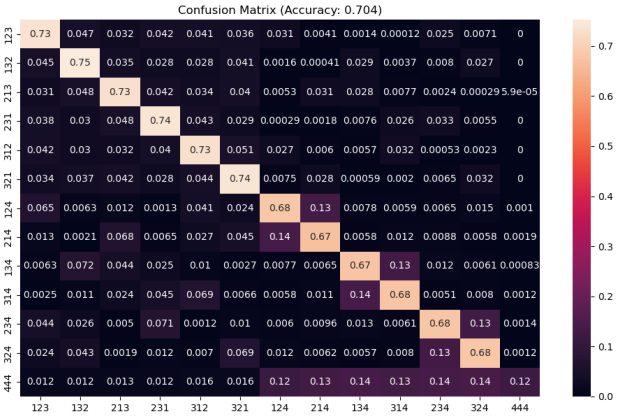


Fig. 8: Confusion matrix on the test set for the best transformer model on the 4.1 million row simulated patient hyperparameter study.

Model	Max Train Acc.	Max Val Acc.	Test Acc.
LSTM	0.73	0.70	0.71
FCN	0.72	0.73	0.73
1D CNN	0.62	0.54	0.55

TABLE X: Additional study grid search results on the 1.1 million row simulated patient dataset.

around 1-3% worse on this search compared to the 4.1 million row data grid search, which may be due to the much smaller dataset size (1.1 million rows compared to 4.1 million rows); in other words, increased dataset size may lead to better models for this specific scenario. The 1D CNN had much worse performance compared to all other models on the patient dataset. Even though it approximated the accuracy results of previous research, it was trained on a much larger dataset [5]. Hence, CNN models may not be able to understand the spatial characteristics of prompt gamma scatterings as well as other models.

VI. Conclusions

This work found that on simulated patient data, the best FCN model achieved a testing accuracy of 74.2% while the best LSTM model achieved a testing accuracy of 73.7%. Both results constitute a significant ∼18% numerical accuracy increase compared to any previous research result on simulated patient data. In addition to greatly improved accuracy on more complex patient data, these models also constitute a 1-2% increase in testing accuracy compared to any previous BRIDE result on the less complex, less realistic, and easier-to-learn water phantom data. This greatly improved model generalizability offers large improvements to actual model usability in a real-world setting. In a clinical workflow, these accuracy increases would translate into more effective classification of scatter type through a loaded model running on a lightweight GPU or CPU framework; thus, a less noisy reconstructed image of the path of the proton beam with respect to the patient body could be quickly produced. With de-noised images, medical professionals can effectively identify Bragg peak location to ensure safe radiotherapy treatment without damage to healthy surrounding tissue.

We introduced 2 novel simulated patient datasets based on actual variable-density patient CT measurements that were up to 8 times larger than simulated patient datasets used in previous work. The effects of the increased dataset size was apparent with significantly boosted model robustness and accuracy. In terms of models, a shallower architecture was found to generally perform better due to less overfitting; however, deeper architectures had significant performance with proper regularization. The larger 4.1 million row dataset led to around 5% better testing accuracy in models compared to the 1.1 million row dataset.

This work introduced several novel implementations of machine learning models in order to develop frameworks that better understood the dependency patterns between scattering events, including the Event Classifier Transformer Neural Network and the 1D CNN. The prior models, along with an FCN and LSTM, were optimized on a large scale using a newly developed BRIDE platform hyperparameter tuning framework.

VII. Future Work

It is important to note that the achieved model accuracies of ∼74% are not currently sufficient for use in a clinical setting. Models are suggested to reach at least around 90% accuracy for clinical usability [7]. Future work for additional model enhancements in terms of the incorporation of more advanced models and architectures to better find patterns within the given data is necessary and beneficial. Specifically, graph neural networks (GNNs) and hierarchical classification may be promising in this area of study.

A. Graph Neural Networks

A possible future direction is implementing graph neural networks, specifically Graph Attention Networks (GAT), to classify the scatter events. Currently, events are inputted as flattened vectors, ignoring their natural spatial and relational structure. However, each triple-hit event can be modeled as a graph with nodes representing hits with spatial and temporal features like position, energy, and time, while edges encode the relationships between the hits. The hits are not independent and their ordering, spacing, and energy differences often contain useful information for classification purposes.

GAT models are suited for this task since they update node features by aggregating information from its neighbors using learned attention weights, allowing the model to prioritize more relevant neighbors [15]. They are then able to capture the dependencies between the hits instead of flattening the data into one vector. This method may have the capability to improve classification accuracy when it comes to the scatter events.

B. Hierarchical Classification

Another direction that can be explored in future research is to replace the current flat classifier with a hierarchical one. The current models map scattering events to one of the 13 classes directly, requiring it to distinguish among all of them during training. However, as noted earlier, of the 13 classes, there consist of 3 distinct groups of classes: true triples, DtoTs, and false triples.

Given this, a possibly more efficient strategy to classify the scatterings could be to use a hierarchical classifier with two main components: a general model that categorizes events into true triples, DtoT, and false triples, followed by three specialized models that further classify true triples, DtoTs, and false triples into their respective subclasses. This approach has the potential to improve both the efficiency and accuracy of the current classification pipeline by reducing the amount of classes at each step [14].

Acknowledgment

This work is supported by the grant "REU Site: Online Interdisciplinary Big Data Analytics in Science and Engineering" from the National Science Foundation (grant no. OAC–2348755). Co-authors Sharma and Ren additionally acknowledge support by NIH. We acknowledge the UMBC High Performance Computing Facility and the financial contributions from NIH, NSF, CIRC, and UMBC for this work.

References

[1] Alina M. Ali, David Lashbrooke, Rodrigo Yepez-Lopez, Sokhna A. York, Carlos A. Barajas, Matthias K. Gobbert, and Jerimy C. Polf. Determining optimal configurations for deep fully connected neural networks to improve image reconstruction in proton radiotherapy. Technical Report HPCF–2021–12, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2021.

[2] Kaelen Baird, Sam Kadel, Brandt Kaufmann, Ruth Obe, Yasmin Soltani, Mostafa Cham, Matthias K. Gobbert, Carlos A. Barajas, Zhuoran Jiang, Vijay R. Sharma, Lei Ren, Stephen W. Peterson, and Jerimy C. Polf. Enhancing real-time imaging for radiotherapy: Leveraging hyperparameter tuning with PyTorch. Technical Report HPCF–2023–12, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2023.

[3] Angelo Calingo, Bikash Gautam, Peter L. Jin, Sidhya Pathak, Michelle Zhao, Hussam Fateen, Harrison Lewis, Matthias K. Gobbert, Vijay R. Sharma, Lei Ren, Ananta Chalise, Stephen W. Peterson, and Jerimy C. Polf. Github repository. https://github.com/big-data-lab-umbc/big-data-reu, 2025.

[4] Angelo Calingo, Bikash Gautam, Peter L. Jin, Sidhya Pathak, Michelle Zhao, Hussam Fateen, Harrison Lewis, Matthias K. Gobbert, Vijay R. Sharma, Lei Ren, Ananta Chalise, Stephen W. Peterson, and Jerimy C. Polf. Large-scale optimizations in proton beam radiotherapy by neural network denoising of robust simulated patient data. Technical Report HPCF–2025–5, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2025.

[5] Michael O. Chen, Julian Hodge, Peter L. Jin, Ella Protz, Elizabeth Wong, Ruth Obe, Ehsan Shakeri, Mostafa Cham, Matthias K. Gobbert, Carlos A. Barajas, Zhuoran Jiang, Vijay R. Sharma, Lei Ren, Sina Mossahebi, Stephen W. Peterson, and Jerimy C. Polf. Using neural networks to sanitize Compton camera simulated data through the BRIDE pipeline for improving gamma imaging in proton therapy on the ada cluster. Technical Report HPCF–2024–5, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2024.

[6] Michael O. Chen, Julian Hodge, Peter L. Jin, Ella Protz, Elizabeth Wong, Ruth Obe, Ehsan Shakeri, Mostafa Cham, Matthias K. Gobbert, Carlos A. Barajas, Vijay R. Sharma, Sina Mossahebi, Lei Ren, Stephen W. Peterson, and Jerimy C. Polf. Improving gamma imaging in proton therapy by sanitizing compton camera simulated patient data using neural networks through the bride pipeline. In *2024 IEEE International Conference on Big Data (Big Data 2024)*, pages 7463–7470, 2024.

[7] Joseph Clark, Anaise Gaillard, Justin Koe, Nithya Navarathna, Daniel J. Kelly, Matthias K. Gobbert, Carlos A. Barajas, and Jerimy C. Polf. Sequence-based models for the classification of Compton camera prompt gamma imaging data for proton radiotherapy on the GPU clusters taki and ada. Technical Report HPCF–2022–12, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2022.

[8] Jonathan R. Hughes and Jason L. Parsons. FLASH radiotherapy: Current knowledge and future insights using proton-beam therapy. *Int. J. Mol. Sci.*, 21(18):6492, 2020.

[9] Daniel W. Mundy and Michael G. Herman. An accelerated threshold-based back-projection algorithm for compton camera image reconstruction. *Medical Physics*, 38(1):15–22, 2011.

[10] Ruth Obe, Brandt Kaufmann, Kaelen Baird, Sam Kadel, Yasmin Soltani, Mostafa Cham, Matthias K. Gobbert, Carlos A. Barajas, Zhuoran Jiang, Vijay R. Sharma, Lei Ren, Stephen W. Peterson, and Jerimy C. Polf. Accelerating real-time imaging for radiotherapy: Leveraging multi-GPU training with PyTorch. In *2023 International Conference on Machine Learning and Applications (ICMLA 2023)*, pages 1735–1742, 2023.

[11] Costanza M. V. Panaino, Ranald I. Mackay, Karen J. Kirkby, and Michael J. Taylor. A new method to reconstruct in 3D the emission position of the prompt gamma rays following proton beam irradiation. *Sci. Rep.*, 9(1):18820, 2019.

[12] Jerimy C Polf, Stephen Avery, Dennis S Mackin, and Sam Beddar. Imaging of prompt gamma rays emitted during delivery of clinical proton beams with a compton camera: feasibility studies for range verification. *Physics in Medicine & Biology*, 60(18):7085, 2015.

[13] Jerimy C. Polf, Carlos A. Barajas, Stephen W. Peterson, Dennis S. Mackin, Sam Beddar, Lei Ren, and Matthias K. Gobbert. Applications of machine learning to improve the clinical viability of Compton camera based in vivo range verification in proton radiotherapy. *Front. Phys.*, 10:838273, 2022.

[14] P. M. Rezende, J. S. Xavier, D. B. Ascher, G. R. Fernandes, and Pires D. E. V. Evaluating hierarchical machine learning approaches to classify biological databases. *Briefings in bioinformatics*, 23(4), 2022.

[15] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.