This item is likely protected under Title 17 of the U.S. Copyright Law. Unless on a Creative Commons license, for uses protected by Copyright Law, contact the copyright holder or the author.

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

Nithya Navarathna

Dr. Matthias K. Gobbert

Department of Computer Science and Electrical Engineering

The Power of GPUs in Machine Learning to Improve Proton Beam Therapy for Cancer Treatment

Biography

Nithya Navarathna is an undergraduate senior at UMBC pursuing a dual degree in biological sciences and bioinformatics and computational biology, with a minor in computer science, and is on the pre-medicine track. She is a Meyerhoff Scholar, Louis Stokes Alliance for Minority Participation (LSAMP) Scholar, and also a member of UMBC's Grand Challenges Scholars Program. She is also a two-time Undergraduate Research Award (URA) recipient. In the future, Nithya hopes to become a physician-scientist by attending an MD/PhD program after her undergraduate studies.

Nithya would like to thank her research mentor, Dr. Matthias K. Gobbert, for his guidance and support throughout this research project. Additionally, she would like to thank her biology research mentor Dr. Achuth Padmanabhan, for their continued support and encouragement.

Research Journey

When I first enrolled in UMBC, I had a strong desire to be involved in STEM research, which was further encouraged and supported by the Meyerhoff Scholars Program. Through participating in the Louis Stokes Alliance for Minority Participation (LSAMP) Research Program my freshman year, I joined The Padmanabhan Lab in the Biological Sciences Department as an undergraduate researcher. To further explore my research interests, I was seeking a summer internship that combined my interests in both biology and computer science for the summer after my junior year. My LSAMP advisor suggested that I apply to the National Science Foundation's Big Data REU 2022 at UMBC. The research project was a great fit because the project's applications were in cancer biology while the actual project was solely computational. Specifically, my project explores how the utilization of graphics processing units (GPUs) can be advantageous when used with machine learning models to improve proton beam therapy for cancer.

Abstract

Proton beam therapy utilizes proton beams to treat cancerous tumors while avoiding unnecessary radiation exposure to surrounding healthy tissues. Real-time imaging of proton beams while they travel through a patient's body can make this form of radiotherapy more precise and safer for the patient. The use of a Compton camera is one proposed method for real-time imaging of the prompt gamma rays that are emitted by proton beams. Unfortunately, some of the Compton camera data is flawed, and the image reconstruction algorithm yields noisy and insufficiently detailed images to evaluate proton delivery for the patient. Machine learning can be a powerful tool to clean up the Compton camera images. Previous work used a deep residual fully connected neural network, but the use of recurrent neural networks (RNNs) has been proposed since they use recurrence relationships to make potentially better predictions. In this work, RNN architectures using two different recurrent layers are tested, the LSTM and the GRU. Although the deep residual fully connected neural network achieves over 75% testing accuracy and our models achieve only over 73% testing accuracy, the simplicity of our RNN models containing only 6 hidden layers as opposed to 512 is a significant advantage. This greatly decreases the amount of time it takes to load the model from the disk, potentially enabling the use of Compton camera image reconstruction in real time during patient treatment. A graphics processing unit (GPU), known to perform complex mathematical calculations to display high-quality graphics, could enable the use of this approach in a clinical setting since they are small and affordable.

1. Introduction

Proton beam therapy has gained popularity as a cancer treatment due to its many advantages. With cancer being the second highest cause of death in the United States, radiation therapies have been widely used as a treatment [9]. Also known as radiotherapy, radiation therapy uses high-energy particles such as x-rays, gamma rays, or protons, to damage the DNA of target cancer cells. X-ray therapy can deliver dosage at the tumor site, and its radiation continues to travel through the body until it exits on the other side. This may potentially harm healthy surrounding tissues and organs that are unnecessarily exposed to radiation. In contrast, proton beams have a finite range that can be controlled, and deposit the majority of their energy just before they stop depositing energy. This sharp energy increase of the proton beam right before stopping is known as the Bragg peak. Since almost no radiation is delivered beyond the Bragg peak, healthy tissue can be spared from unnecessary radiation [9]. Thus, the Bragg peak allows proton therapy to be advantageous in delivering the radiation dosage directly at the tumor site without traveling further into the body.

To take full advantage of the properties of proton therapy, we must have an efficient technique to image the prompt gamma rays produced by the beam in real-time, as they travel through the patient's body. A Compton camera is one instrument that can be used to detect the prompt gamma rays emitted when the proton beam travels through the body. Moreover, an algorithm is available to reconstruct the beam's image from the prompt gamma data, which then provides an indirect image of the proton beam. Unfortunately, a lot of the raw data of the Compton camera is flawed, and the reconstruction algorithm yields noisy and insufficiently detailed images to evaluate the proton delivery for the patient [7, 8].

Machine learning can be used to clean the raw Compton camera data by identifying and removing false data before image reconstruction [7,8]. Research efforts to clean the Compton camera data have led to the use of neural networks. Shallow networks like the one in [7] use 1 to 2 hidden layers to perform simple classifications of simulated prompt gamma data under ideal conditions that do not represent the irradiation conditions encountered during clinical proton beam radiotherapy. This shallow network in [7] is a binary classification network that simply determines which event data are true events and should be used for reconstruction and which are false events that should not be used for reconstruction. This is improved upon in [8] by using the deep residual fully connected neural network described in [3] for triple event classification. This neural network consists of 64 residual blocks with 8 fully connected layers per block yielding a total of 512 hidden layers. Each layer had 256 neurons per layer, a 45% dropout rate, and used leaky ReLU activation. More detailed results and discussions about the impact of neural network

processing on the use and viability of Compton camera-based imaging in clinical proton radiotherapy are the focus of [8], while providing details on the network and its training are the focus of [3]. The full capabilities of the described neural network are specified in [2], where preprocessing the data, all classification capabilities, and postprocessing output data are described in detail. Other recent work [1, 10] focused on hyperparameter studies on the deep residual fully connected neural network from [3], varying batch sizes, neurons, and layers. The use of recurrent neural networks (RNNs) is proposed in [1] since they use recurrence relationships in sequence data sets to make potentially better predictions. The potential for RNNs to be an improvement over feedforward neural networks (FNNs) is shown in [6].

In this work, we test RNN architectures using two different recurrent layers because of their potential for classifying sequence data, the Long Short-term Memory (LSTM) (discussed in **Section 3.1**) and the Gated Recurrent Unit (GRU) (discussed in **Section 3.2**). The LSTM uses memory cells with gates and a carry track to encode and learn from sequence data. The GRU uses two gating units to encode and learn from sequence data. The goal of this change in the type of network architecture is to examine data as a sequence of interactions rather than one single event, but preliminary results do not show any benefit. We use models with 4 GRU layers and with 4 LSTM layers and achieve similar testing accuracy as the deep residual fully connected model from [3]. The model with 4 GRU layers outperforms the deep residual fully connected model in 3 classes but has a larger gap (within 10%) in accuracy in the other 10 classes. The model with 4 LSTM layers outperforms the previous deep residual fully connected model in only 2 classes but has a smaller gap (within 6%) in accuracy in the other 11 classes. Although the deep residual fully connected model achieves slightly higher accuracy in nearly every class, the simplicity of our RNN models containing only 6 hidden layers (4 recurrent and 2 fully connected) as opposed to 512 is an advantage. And importantly in a clinical setting, this advantage could enable the use of real-time Compton camera image reconstruction during patient treatment.

A graphics processing unit (GPU), known to perform complex mathematical calculations to display high-quality graphics, could enable the use of this approach in a clinical setting since they are small in size and affordable. With this motivation, we use the available GPU partitions in the UMBC High Performance Computing Facility (hpcf.umbc.edu) to test and compare their performance for this application problem. HPCF has several GPU partitions in the clusters taki and ada. The taki system has two GPU partitions 2013 and 2018. For 2018, This 1 GPU node has four NVIDIA Tesla V100 GPUs (5120 computational cores over 84 SMs, 16 GB onboard memory) connected by NVLink, two 18-core Intel Skylake CPUs, and 384 GB of memory (12 × 32 GB DDR4 at 2666

MT/s). The 2013 GPU node contains 18 hybrid CPU/GPU nodes, each with two NVIDIA K20 GPUs (2496 computational cores over 13 SMs, 4 GB onboard memory), two 8-core Intel E5-2650v2 Ivy Bridge CPUs (2.6 GHz clock speed, 20 MB L3 cache, 4 memory channels), and 64 GB of memory (8×8 GB DDR3). Networks built on Taki were built using Tensorflow v2.4.0 (www.tensorflow.org) with the bundled Keras module. We also used SciKit-learn v0.23.dev0 (https://scikit-learn.org/stable/) to preprocess and normalize the data. Moreover, pandas v1.1.0 (https://pandas.pydata.org/) and NumPy v1.18.1 (www.numpy.org) were also used to help preprocess the data. Finally, we used the Matplotlib v3.5.1 (www.matplotlib.org) library to graph our results.

The ada system has 3 distinct node types. Four nodes each with 8 Nvidia RTX 2080 Ti GPUs each with 11GB GPU memory. Seven nodes with 8 Nvidia Quadro RTX 6000 GPUs each with 24GB of GPU memory. Two nodes each with 8x Nvidia Quadro RTX 8000 GPUs each with 48GB memory. Each node has 384 GB of CPU memory (12×32 GB DDR4 at 2933 MT/s) except the two RTX 8000 nodes which have 768GB of CPU memory(12×64 GB DDR4 at 2933 MT/s). Networks built on ada were built with the software package Anaconda3 and Tensorflow v2.6.0 with the bundled Keras module.

The remainder of this report is organized as follows: **Section 2** provides the background on proton beam therapy to treat cancer and the Compton camera to image prompt gamma rays. **Section 3** details the basics of machine learning and recurrent neural networks, while also providing details on the LSTM and GRU. **Section 4** contains selected application-oriented results using our trained network, while **Section 5** presents the performance results using the GPUs described above.

2. Application Background

2.1 Proton Beam Therapy

Radiation therapy is a form of cancer treatment that uses high doses of radiation to kill cancer cells. X-ray therapy, a form of radiation therapy, is a common technique used for cancer treatment, where the majority of the radiation dosage is delivered upon entering the body. Because of this, the tumor does not receive as high of a concentrated dose as it should. In addition, X-rays will continue to travel posteriorly into the human body until it exits on the other side. This is not ideal as there is no need for extra radiation exposure within the body. Proton therapy on the other hand, which is another form of radiation therapy, is more efficient in this manner. Rather than depositing the majority of the dosage at the entry site, proton therapy works to deposit the majority of the dosage at the tumor site itself, thus making the process more effective. Proton therapy also has an advantage over X-ray therapy in the sense that the proton

beam travels no further posterior into the body than the site of the tumor, allowing for minimal exposure to surrounding tissue. To fully take advantage of all the benefits that proton therapy has to offer, we must have a sufficient technique to monitor the proton beam's path in real time as it travels through the patient's body.

When delivering a dosage to a tumor, the professional treating the patient will create a safety margin that enlarges the treatment area to ensure that all parts of the tumor are guaranteed to receive dosage. The safety margin is needed to account for slight movements in the patient during treatment as well as slightly different positioning of the patient from one treatment to the next, over several weeks. The availability of real-time information on the trajectory of the proton beam through the patient's body during treatment could enable us to make the safety margin smaller and use the optimal path. The use of Compton cameras is one proposed method for the real-time imaging of prompt gamma rays that are emitted by the proton beams as they travel through the body.

2.2 Compton Camera

The Compton camera is a multi-stage detector that produces data used to generate images of proton beams used in proton beam therapy [3]. As protons from the beam enter the body, they interact with cells in the body causing the emission of prompt gamma rays. Some of these gamma rays will collide with the Compton camera. An interaction is when a prompt gamma collides with a stage of the Compton camera. For each interaction, the camera records x-, y-, z-coordinates and the energy level of the scatter. The readout of interactions in a single period is called an event. The raw output data from the camera for each interaction is in the form (e_i, x_i, y_i, z_i) where $i = 1, 2, 3$ for the three stages of the Compton camera, and e_i is the energy level.

Image reconstruction algorithms exist that can recover the path of the proton beam from the Compton camera data. The Compton camera's capability to reconstruct full 3D images of the proton beam range could be used with the patient's CT scan to compare the planned treatment dose and make adjustments. Radiotherapy treatment requires conformity between the treatment plan and the treatment delivery, making sure that the patient's bone and soft tissue landmarks are aligned as they were at the time of treatment planning [9]. If a patient changes position, such as wiggling, scratching, looking the other way, or doing any other subtle movement, it could disrupt the treatment plan. By obtaining reliable information regarding the patient from the reconstructed images, clinicians have the opportunity to better ensure that the entire tumor receives the exact dose as planned while making sure surrounding healthy tissues are safe.

Although the Compton camera is able to detect interactions, prompt gammas are emitted close to the speed of light and, consequently, the camera is unable to detect the true ordering of interactions in an event. The false events that do not detect the true ordering of interactions cause noise in the image created, impacting the usefulness of the image [3]. There are three different types of scatters detected by the Compton camera:

a. **True Triples:** In the True Triples event, the Compton camera will detect the path of the prompt gamma. However, the true path may be some other ordering. There are a total of 6 total combinations of True Triple scatters: 123, 132, 213, 231, 312, 321 and, as the data stands, only the 123 ordering is usable.

b. **Double-to-Triples (DtoT):** In the DtoT event, the Compton camera will detect the path of a single prompt gamma as a true triple. However, in reality, two prompt gammas had varying paths. One prompt gamma could have been detected as the first and third interaction and the second prompt gamma could have been mistaken as the second interaction. Similar to true triples, there are a total of 6 misdetection orderings: 124, 134, 214, 234, 324, 314. The second prompt gamma interaction is classified as "4" in the misdetection orderings. In this case, without processing the data, all 6 orderings are unusable.

c. **False Triples:** In a false triples event, the Compton camera will detect a true triple whereas, in reality, there were three different prompt gammas. As a result, this entire event provides no insight into the path of a single prompt gamma and must be discarded.

2.3 The Need for Machine Learning

To make proton beam therapy more effective, real-time imaging is needed to verify the location and effectiveness of the proton beam, in particular the location of the Bragg peak. Machine learning is capable of classifying which type of scatter event occurred based on data provided by the Compton camera. These classifications lead to the removal of unusable data which will clean the resulting image. A clearer image allows for treatment verification. A sufficiently accurate machine learning model could produce an image that is clear enough to be used in proton beam therapy as a form of treatment verification. A machine learning algorithm will need approximately 90% testing accuracy to be useful for clinicians.

3. Machine Learning

Machine learning is a type of artificial intelligence where a machine is trained to identify specific trends and patterns to make predictions from data. In the case of Compton camera data, the machine learning algorithm will try to predict the appropriate class for a scatter event. The main benefit of machine learning is its efficiency in producing results that would take humans alone much longer to produce. There are four different ways that a machine can be taught: supervised, unsupervised, semi-supervised, and reinforcement. Supervised learning is a form of learning where the machine is provided a labeled data set that has regular input data as well as the desired output data. This allows the machine to produce a model that has been fitted appropriately. Unsupervised learning is used when one wants to identify hidden patterns within an unlabeled data set. This allows the machine to identify any trends it finds in the data without special instruction. Semi-supervised learning is a mixture of supervised and unsupervised where the model is provided some labeled data and a large amount of unlabeled data. Reinforcement learning is similar to the way humans learn where the machine will interact with the data and there will be either a positive or negative reward depending on whether the machine does something the programmer wants or not. The method used in this study is supervised learning because the data set contains both the data from the scatter event and the corresponding label of which event scatter took place.

Recurrent neural networks (RNNs) are efficient neural networks used for time series tasks. They work similarly to a coupling process in biology. They rely on information from the previous system or "loop" to move forward with the next. In this type of neural network, the sequence or order of the network is very important. The system can be read and executed differently if the elements of both series are in different orders. In the case of RNNs, elements include an input layer, hidden layers, and an output layer. RNNs use back-propagation through time to illustrate gradients. The difference between RNN backpropagation and other methods such as in a feed-forward network is that sum errors are necessary at each time step because of the shared parameters throughout the network. Several types of RNNs are distinguished by the pathways between inputs and outputs. RNNs may also contain activation functions that allow a neuron to translate the input into a specific output. Finally, there are a few RNN structures that vary depending on the desired use. There are bidirectional recurrent neural networks, long short-term memory, and gated recurrent units. Bidirectional recurrent networks rely on future data to generate predictions.

RNNs are a viable option for Compton camera data because of their ability to encode information about previous events. Shaping an event in the Compton camera as a sequence of three interactions each with five

features, we have transformed the data produced by the Compton camera into a sequence. Using the sequence of interactions, the RNN will be able to predict the ordering of interactions, i.e., the appropriate scatter.

3.1 Long Short-Term Memory

A Long Short-Term Memory (LSTM) neural network is a type of RNN that is traditionally used for natural language processing and time series forecasting. The unique aspect of LSTM is that it contains a memory cell. This memory cell is used to store certain pieces of information that may be needed later in the training process, called a state. The memory cell has three gates to determine the state: forget gate, input gate, and output gate. The forget gate controls what stored information can be forgotten. The input gate controls what information should be used to change the state of the memory cell, and the output gate controls which part of that information is needed at a given time. As stated previously, RNNs use the output of one step and carry it over into the next step in addition to the new data input. The memory cell was added to combat the main issue with RNNs which is long-term dependency whereas more and more information is fed into the RNN, it becomes less effective in learning because the network cannot remember everything.

3.2 Gated Recurrent Unit

A Gated Recurrent Unit (GRU) is essentially a streamlined version of the LSTM in **Section 3.1**. The GRU has gating units that modulate the flow of information inside the unit. The GRU factors in the previous short-term dependency with a reset gate by using a linear interpolation between the previous activation function value and the current one. The GRU also factors in previous long-term dependencies with an update gate by taking a linear sum between the existing state and the newly computed state. Unlike the LSTM the GRU does not have separate memory cells.

4. Machine Learning for Compton Camera Imaging

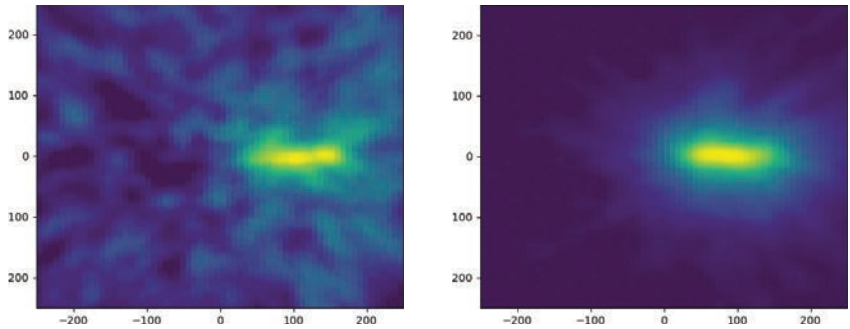
For our studies, we trained the neural network on a data set that was generated using a Monte Carlo simulation and that consisted of 1,443,993 records and 15 features. These features represent each interaction's spatial coordinates, Euclidean distance, and energy deposition. An interaction is a grouping of three spatial coordinates and an energy level. Each row is either a triple, double-to-triple or a false triple and consists of three interactions each. Our training data set only consisted of True Triples, Double-to-Triple scatter, and False events. Furthermore, when

testing the neural network, we used datasets that used 150MeV (Mega electron Volt) beams with three different dosage rates: 20kMU (kilo Monitor Unit), 100kMU, and 180kMU. The larger kMU values correspond to more intense dosage rates. Both the training and testing datasets were reshaped to be sequentially read. Therefore, each record of 15 features was reshaped to 3 interactions of 5 features each: three spatial coordinates, Euclidean distance, and energy deposition. Each record is fed into the neural network as a sequence of 3 interactions. The testing data contains 37,151 testing data points for 20kMU/min, 17,425 for 100kMU/min, and 12,254 for 180kMU/min from MCDE model test 1 150MeV. More details on these studies and results are available in [4, 5].

Model	**Accuracy**	**Load Time**
DRFCN (512 FCL)	75.8%	47s
1 LSTM, 256 FCL	74.6%	24s
4 LSTM w/ more neurons	74.4%	15s
2 LSTM, 128 FCL	74.2%	13s
4 LSTM, 64 FC	70.0%	11s
4 GRU	73.4%	10s
4 LSTM	73.2%	7s

Table 1: Comparison of top-performing models with the deep residual fully connected network (DRFCN) from [3].

The key results of our work are summarized in **Table 1** and are detailed in [5]. The Model column refers to the architecture of the model used. The first row shows the results of the deep residual fully connected network (DRFCN) in [2]; this model has 512 fully connected layers (FCL). All of the following rows correspond to the various models tested in [5] while 4 GRU represents the model with 4 GRU layers and 2 dense layers of 128 and 64 neurons. 4 LSTM represents the model with 4 LSTM layers and 2 dense layers of 128 and 64 neurons. The Accuracy column represents the overall testing accuracy of the model at the dosage rate of 100kMU/min. The Load Time column represents the observed wall clock time in seconds to load the model from its saved state to an active state, i.e., from disk to GPU memory.



a. 20kMU/min Uncleaned **b.** 20kMU/min Uncleaned

Figure 1: Image **(a)** uses testing data without the NN classification for data correction, called "uncleaned" data. Image **(b)** uses testing data with NN classification for data correction, called the "cleaned" data with the 4-layer LSTM model. Testing data used comes from MCDE model test1 150MeV. Results for additional treatment regimes are included in [4].

The DRFCN model has the highest accuracy of 75.8% with a load time of 47s. The models in the last two rows of the table have accuracies of 73.4% and 73.2% respectively while loading in 10s and 7s. These 4 GRU and 4 LSTM models are much simpler with only 6 hidden layers instead of 512. In particular, they have a factor of 85 fewer layers while being only 2% less accurate. These two recurrent models are also 4 times faster to disk an advantage when treating the patient.

To illustrate the effect that network event classification can have on the PG images produced from the camera data, reconstructed PG images are shown in **Figure 1**. The image in the left column is the PG image reconstructed with raw data before neural network classification, called the "uncleaned" data. The image in the right column is the PG image reconstructed with data after it has been corrected based on the neural network classifications, called the "cleaned" data. Since each PG image is from data collected during delivery of the same 150MeV proton beam, they will have the same position and range even though they are reconstructed from data collected at different dose rates. We observed an improved visual appearance of the beam in which the start point and end point are now easily distinguishable at all three dose rates. The method used to reconstruct these images is described in [3].

5. GPU Performance Tests on Taki and Ada Partitions

The tests in **Table 2** were performed to compare the performances of the 2013 taki partition with the 2018 taki partition, along with the three GPUs of ada- RTX 2080 Ti, RTX 6000, and RTX 8000. All studies in the performance comparison study were run using a deep fully connected neural network whose architecture is similar to the model in [3] with residual blocks and some hyperparameter changes. The hyperparameters used for these tests include 128 layers with 256 neurons, a batch size of 8192, a learning rate of 1e-3, and varying epochs from 64 to 1024 epochs. **Table 2** record the performance times for each partition based on the number of epochs. The taki 2018 partition performs the fastest completing the job in 4 hours, 13 minutes, and 42 seconds for 1024 epochs. The slowest performance is that of the taki 2013 partition, which takes 15 hours, 49 minutes, and 31 seconds for the same number of epochs. All three ada partitions perform similarly and are slightly slower than the taki 2018 partition. The taki 2018 partition is at least three times faster when compared to the taki 2013 partition and is the most efficient partition to use for future studies.

In **Table 2**, the taki 2018 GPU cluster was shown to have the fastest GPU node. The performance of the GPU nodes on ada is very similar to those on the taki 2018 GPU, but ada has many more available GPUs. There are 56 RTX 6000 GPUs available and only 4 GPUs available on taki 2018. The taki 2013 GPUs are too slow for the studies in this research. The number of high-performance GPUs on ada is a huge advantage for performing numerous simulations simultaneously.

The Load Time measurements from **Table 1** report observations on a reference computer, a basic laptop with an 11th Gen Intel Core i7–1165G7 CPU at 2.80 GHz with 16 GB of memory. The laptop has Intel Optane Memory H10 with 512 GB Intel QLC 3D NAND solid state drive connected by PCIe 3.0 x4 with NVMe interface. The GPU on the laptop is an Intel Iris Xe Graphics card. On a large cluster like taki or ada, described at the end of **Section 1**, these times would be slower, since the central rotating disk storage is much larger and connected only via network cables to the compute nodes. Even with high-performance fiber-optic cables, this is slower than a direct connection from solid-state storage inside a laptop. However, such direct connection and the use of solid-state storage are more realistic for the type of computer used in a clinical setting in a treatment room. The use of a GPU in the treatment laboratory can significantly decrease the load times, and it is a realistic possibility since GPUs are small, affordable, and can easily fit in the treatment room. More details on these studies and results are available in [4].

a. Performances with 64 epochs

Cluster	Partition	Total Time (hh:mm:ss)
taki	2013	00:59:49
taki	2018	00:16:01
ada	RTX 2080	00:19:17
ada	RTX 6000	00:20:32
ada	RTX 8000	00:21:08

b. Performances with 128 epochs

Cluster	Partition	Total Time (hh:mm:ss)
taki	2013	01:59:06
taki	2018	00:31:48
ada	RTX 2080	00:39:16
ada	RTX 6000	00:40:05
ada	RTX 8000	00:41:06

c. Performances with 256 epochs

Cluster	Partition	Total Time (hh:mm:ss)
taki	2013	04:57:11
taki	2018	01:03:21
ada	RTX 2080	01:18:38
ada	RTX 6000	01:20:04
ada	RTX 8000	01:20:17

d. Performances with 512 epochs

Cluster	Partition	Total Time (hh:mm:ss)
taki	2013	10:30:07
taki	2018	02:06:39
ada	RTX 2080	02:38:19
ada	RTX 6000	02:40:09
ada	RTX 8000	02:41:34

e. Performances with 1024 epochs

Cluster	Partition	Total Time (hh:mm:ss)
taki	2013	15:49:31
taki	2018	04:13:42
ada	RTX 2080	05:09:07
ada	RTX 6000	05:18:45
ada	RTX 8000	05:25:41

Table 2: Table of taki and ada performances with varying epochs.

Acknowledgments

This work would not have been possible without my teammates (Joseph Clark, Anaise Gaillard, and Justin Koe) in the Big Data REU Site 2022 at UMBC (bigdatareu.umbc.edu). Additionally, I would like to thank our team's research assistant Daniel J. Kelly and collaborators Dr. Carlos A. Barajas (Department of Mathematics and Statistics, UMBC) and Dr. Jerimy C. Polf (Department of Radiation Oncology, University of Maryland School of Medicine). 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–2050943). The hardware used in the computational studies is part of the UMBC High Performance Computing Facility (HPCF). The facility is supported by the U.S. National Science Foundation through the MRI program (grant nos. CNS–0821258, CNS–1228778, OAC–1726023, and CNS—1920079) and the SCREMS program (grant no. DMS–0821311), with additional substantial support from the University of Maryland, Baltimore County (UMBC). See hpcf.umbc.edu for more information on HPCF and the projects using its resources.

References

1. Alina M. Ali, David Lashbrooke, Rodrigo Yepez-Lopez, Sokhna A. York, Carlos A. Barajas, Matthias K. Gobbert, and Jerimy C. Polf. Towards 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. Carlos A. Barajas. *Neural Networks for the Sanitization of Compton Camera Based Prompt Gamma Imaging Data for Proton Radiotherapy*. Ph.D. Thesis, Department of Mathematics and Statistics, University of Maryland, Baltimore County, 2022.

3. Carlos A. Barajas, Matthias K. Gobbert, and Jerimy C. Polf. Deep Residual Fully Connected Neural Network Classification of Compton Camera Based Prompt Gamma Imaging for Proton Radiotherapy. *Front. Phys.*, 11:903929, 2023.

4. Joseph Clark, Anaise Gaillard, Justin Koe, Nithya Navarathna, Daniel J. Kelly, Matthias K. Gobbert, Carlos A. Barajas, and Jerimy C. Polf. Multi-layer recurrent neural networks for the classification of Compton camera based imaging data for proton beam cancer treatment. In *9th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT 2022)*, 2022.

5. 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.

6. Fengdi Guo. Comparison of feedforward and recurrent neural networks for predicting pavement roughness, 2021.

7. Enrique Muñoz, Ana Ros, Marina Borja-Lloret, John Barrio, Peter Dendooven, Josep F. Oliver, Ikechi Ozoemelam, Jorge Roser, and Gabriela Llosá. Proton range verification with MACACO II Compton camera enhanced by a neural network for event selection. *Sci. Rep.*, 11(1):9325, 2021.

8. 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.

9. Jerimy C. Polf and Katia Parodi. Imaging particle beams for cancer treatment. *Phys. Today*, 68(10):28–33, 2015.

10. Sokhna A. York, Alina M. Ali, David C. Lashbrooke Jr, Rodrigo Yepez-Lopez, Carlos A. Barajas, Matthias K. Gobbert, and Jerimy C. Polf. Promising hyperparameter configurations fordeep fully connected neural networks to improve image reconstruction in proton radiotherapy In *2021 IEEE International Conference on Big Data (Big Data)*, pages 5648–5657, 2021.