ABSTRACT

Title of Thesis: GANs for Simulating Particle Showers in the CLAS12 Electromagnetic Calorimeter

Degree Candidate: Adam Lincoln Bates

Degree and Year: M.S. in Applied Physics and Computer Science, 2021

Thesis Directed By: William B. Phelps, Ph.D

Generative adversarial neural networks (GANs) are a new form of deep learning model with a variety of applications that still need to be explored. A GAN is a deep learning model in which two neural networks, a generator and a discriminator, contest in an attempt to learn the patterns given by the data accurately and efficiently. Since their creation, GANs have rapidly gained attention in machine learning literature with many applications in natural image processing. At the Thomas Jefferson National Accelerator Facility or Jefferson Lab, the CLAS12 detector is part of the larger energy-doubling project of Jefferson Lab's Continuous Electron Beam Accelerator Facility, and began taking data in 2017. Situated in Hall B, it is used to study electro-induced nuclear and hadronic reactions. A deep learning model to enable high-fidelity fast simulation of particle showers in electromagnetic calorimeters can be applied to the CLAS12 system. If a GAN is applied to the CLAS12 system, it could provide accurate simulations while massively improving efficiency. Unthinkable amounts of simulation could be achieved years before it would otherwise be computationally feasible.

GANS FOR SIMULATING PARTICLE SHOWERS IN THE CLAS12 ELECTROMAGNETIC CALORIMETER

by

Adam Lincoln Bates

Thesis submitted to the Graduate Faculty of
Christopher Newport University in partial
fulfillment of the requirements
for the degree of
M.S. in Applied Physics and Computer Science
2021

Approved:

William B. Phelps, Chair ______________________________

Dr. David Conner ______________________________

Dr. David Heddle ______________________________

DEDICATION

I want to dedicate this work to those that have helped me over the last four years. I want to thank my family for their continuous support and for making attending college possible. Also, I would like to thank Jefferson Lab and their IT department for giving me my first real job opportunity. Finally, I would not be where I am today without all of the teachers who helped me further my understanding of Computer Science during college.

ACKNOWLEDGEMENTS

I want to acknowledge the Physics, Computer Science, and Engineering department at CNU for providing me with a substantive education. Additionally, I want to thank my mentor, Dr. William Phelps, along with my two committee chairs, Dr. David Heddle and Dr. David Conner. I want to thank Jessica Perry and the IT division staff at Jefferson Lab for introducing me to its wonderful scientific community. Finally, I want to thank all of the Hall B staff at Jefferson Lab for the remarkable work that they have accomplished.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Number Page

CHAPTER I:
ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, AND DEEP LEARNING

Artificial Intelligence

The field of Artificial Intelligence is broad and encompasses both Machine Learning and Deep Learning. AI can be defined as the science and engineering of making machines that can complete a task using some degree of ”intelligence.” This does not mean human intelligence, but rather the computational aspect of the ability to achieve goals in the world. [**Poole**].

History

Artificial Intelligence (AI) studies began in the 1950s, and ever since, the field has made substantial technical advances and shown tremendous potential [**Chollet**]. Until the late 1980s, an approach known as *symbolic AI* was dominant; furthermore, this approach involves handcrafting a sufficient set of explicit rules for manipulating knowledge [**Chollet**]. The early versions of Artificial Intelligence consisted of hardcoded rules created by programmers, which does not count as machine learning. There were attempts in the 1970s and the 1980s to reduce the computational requirements posed by AI, and this was done using expert systems, which use rule based, frame based, and logic based rules to accomplish what can be considered the first successful implementation of AI [**Mueller**]. There were two extended periods of inactivity during AI's development that lasted from the late 1950s to mid 1990s, and this was largely due to criticism, misunderstanding, and lack of funding. In more recent times, there has been a resurgence for AI and Machine Learning due to better computing hardware and new applications. [**Mueller**].

Concept

The goal of AI is to mimic and execute tasks that typically require human intelligence, such as learning, reasoning, and perception. Discerning the word intelligence is important because it does not necessarily mean AI can accomplish the same as a human; rather, that AI is able to use computation to simulate human intelligence to achieve an objective. It is able to simulate different tasks better than others. For example, logical-mathematical, visual-spatial, and bodily-kinesthetic types of tasks perform well using AI, while creative, interpersonal, intrapersonal, and linguistic tasks it struggles with or is arguably unable to mimic [**Mueller**]. It is also dangerous to use the word simulate because AI is not entirely about simulating human intelligence; instead, it is about studying the problems the world presents to intelligence. Researchers frequently use methods that are not observed in people and would require much more computing than physically possible [**McCarthy**].

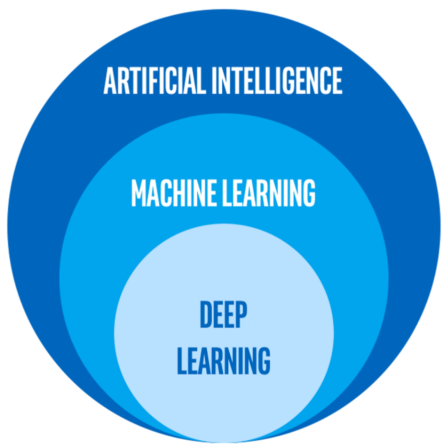


Figure 1: Diagram showing the relationship between AI, machine learning, and deep learning [**Robins**]

Machine Learning

Machine learning is a subfield of artificial intelligence, and it is involved with providing machines with data that they need to learn how to achieve a task without being explicitly programmed to do it. The word learn is important to distinguish because it is not learning in the way a human does, but instead is behaving based on the patterns observed in the data that is fed to the algorithm [**Mueller**].

History

Conceptually, machine learning has been around as long as the idea of a machine capable of learning and originality. The root of machine learning comes from asking if a computer can go beyond what we know how to command it to perform. Machine learning systems are not explicitly programmed, instead they are trained. Practical machine learning did not begin to take off until the 1990s; however, it has now become the most popular and successful subfield of Artificial Intelligence [**Chollet**]. Increases in hardware speed and amounts of data fuels the interest. The field is closely related to mathematical statistics, but differs by exceeding the limits of classical statistical analysis because it deals with large, complex data sets. Machine and deep learning are engineering orientated, and their ideas are proven empirically more often than theoretically [**Chollet**].

Concept

To understand what makes deep learning different from machine-learning approaches, it is important to understand what the machine-learning algorithm actually does. Machine learning can be divided into two categories, supervised and unsupervised learning. Supervised machine learning takes three things: input data, examples of the expected output, and a way to measure whether or not the algorithm is doing a good job. In unsupervised learning, the algorithm learns patterns from unlabeled data without the use of human supervision. The input data is meaningfully transformed to reach closer to the expected output; additionally, the algorithm must be able to gauge accuracy, then adjust to obtain the desired result. To gauge accuracy, neural networks use libraries to minimize a loss function. Weights are assigned to the node along with a bias value. The weights indicate whether or not an algorithm is making good predictions. After training, weights are updated using the backpropogation of error algorithm. The loss is used to calculate the gradients and then the gradients are used to update the weights of the neural network. The learning in machine learning comes from describing an automatic search process for better data representations. Certain data transformations are more useful than others, and machine-learning algorithms are unimaginative when it comes to finding the best transformation. It will simply search through a predefined set of operations known as a *hypothesis space*. This lack of awareness is what differentiates machine-learning from deep learning [**Chollet**].

Deep Learning

Deep Learning is a specific subfield of machine learning. It can be defined as a machine learning technique that creates artificial neural networks to mimic the function of the human brain; specifically, in processing data for use in detecting objects, recognizing speech, translating languages, and making decisions. The more powerful and complex that artificial neural networks become, the ability for deep learning to produce AI and facilitate robust machine learning increases [**Chollet**].

Concept

Deep learning is a method of learning representations from data that emphasizes learning successive layers of increasingly meaningful representations. This successive layers of representation concept provides the *deep* in *deep* learning [**Chollet**]. There are two types of deep learning, supervised and unsupervised. In supervised learning, the neural network is trained using labeled data. In unsupervised learning, the neural network uses unlabeled data and looks for recurring patterns [**Chollet**].

An example of supervised learning would be image classification. If a machine learning algorithm wants to determine whether a picture is of a dog or a cat, it would need data relating to the features of both animals. Human supervision would allow categorization, by the user confirming or denying whether an image is a cat or a dog [**Yagcioglu**].

An example of unsupervised learning would be clustering. If a machine learning algorithm wants to determine the best marketing approach for customers, it can determine patterns through large amounts of data. No human supervision is needed for the algorithm to group similar features together; instead, it finds unique patterns by grouping customers based on similar features found in the data [**Yagcioglu**].

Convolutional Neural Networks

A specific class of deep learning models is a convolutional neural network (CNN). CNNs are commonly used for analyzing visual imagery. Their functionality consists of regularizing versions of multilayer perceptrons. Regularizing means that information is added to solve or prevent problems like overfitting. Multilayer perceptrons are fully connected networks, but full connection can cause the networks to be susceptible to overfitting. To solve this during regularization, CNNs take advantage of the hierarchical pattern in data and assemble more sophisticated patterns using smaller and simpler patterns. Methods to further regularize include assigning and updating weights to the loss function [**Valueva**].

CHAPTER II:
GANS NEURAL NETWORK

Introduction

A generative adversarial network (GAN) is a deep learning model in which two neural networks, a generator and a discriminator, contest in an attempt to learn the patterns given by the data accurately and efficiently. The generator is typically a deconvolutional neural network, while the discriminator is a convolutional neural network [**Karpathy**].

History

The first generative adversarial network was created by Ian Goodfellow and his colleagues in 2014. Deep generative models have difficulty approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies, and with leveraging the benefits of piecewise linear units in the generative context. GANs was created to avoid those issues; additionally, Noise Contrastive Estimation, which uses the same loss function provided motivation for developping GANs [**Goodfellow˙GANs**]. Noise constrastive estimation is a way of learning a data distribution by comparing it against a noise distribution that is defined by the user[**Goodfellow˙GANs**]. Since their creation, Generative adversarial networks have rapidly gained attention in machine learning literature with many applications in natural image processing; however, only a few applications in basic science have been found. Up until *CALOGAN* in 2018, no applications in high energy physics and nuclear physics had been discovered [**Paganini**]. GANs may struggle by generalizing poorly, and suffer what is known as "Mode collapse." This is when the GAN that is being trained misses entire modes from the input data; furthermore, some researchers blame this problem on a weak discriminative network that fails to notice the pattern of omission, while others believe it is a bad choice of objective function [**Goodfellow˙GANs**]. Since its creation, various solutions have built upon Ian's initial concept of a generative adversarial network. Feature matching specifies a new objective for the generator that prevents it from overtraining on the current discriminator, and addresses part of GANs' instability. Another failure of early GANs was that the generator could collapse to a parameter setting in which it always emits the same point. Minibatch discrimination looks at multiple examples in combination, rather than in isolation, and potentially avoids collapse of the generator. Another example is batch normalization, which greatly improves the optimization of neural networks and is highly effective for DCGANs. It can cause an input example to be dependent on several other inputs in the same minibatch; however, another improvement known as virtual batch normalization (VBN) solves this problem by having a reference batch of examples that are chosen once and fixed at the start of training and on the input itself. There are various other optimizations that can be applied to GANs, depending on where the neural network is being implemented [**Salimans**].

Concept

The GANs neural network is a new model for deep neural networks that estimates generative models through an adversarial process. During this process, two models are simultaneously trained: a generative model that captures the data distribution, and a discriminative model that estimates the probability that a sample is real or came from the training data [**Goodfellow˙GANs**]. The generative network is named as such because the network creates candidates using randomized input. It learns to map from a latent space to a data distribution of interest. During training, its objective is to increase the error rate of the discriminative network; in other words, the generator evaluates its success based on its ability to fool the discriminator [**Goodfellow˙GANs**]. The discriminative network is named as such because it differentiates between real and fake distributions by distinguishing candidates produced by the generator from the true data distribution. During training, its objective is to minimize the number of times it is fooled by the discriminator [**Goodfellow˙GANs**]. Training this neural network involves maximizing the probability that the discriminative model will make a mistake. It can be thought of as a minimax two-player game, the generator wants to maximize the mistakes the discriminator makes, and the discriminator wants to minimize its errors [**Goodfellow˙GANs**].

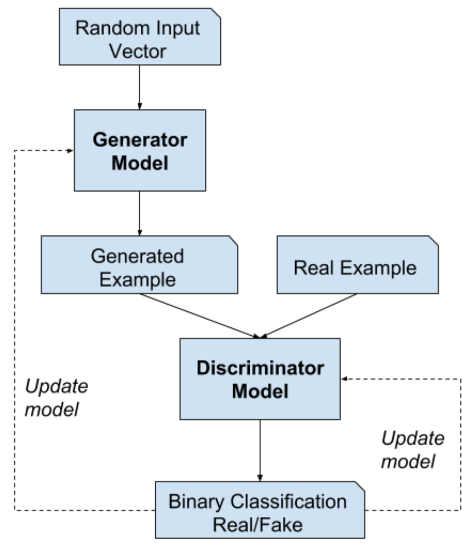


Figure 2: Example of the basic Generative Adversarial Neural Net structure [**Brownlee**]

Utility

The GAN technology excels in using data provided through images to create similar images that follow the same rules; moreover, having this ability creates opportunity in numerous fields. In the arts, GANs can be used to create photographs, or generate photos of realistic looking fashion models [**Yu**]. In science, GANs can improve astronomical images and simulate gravitational lensing for dark matter research [**Schawinski**] [**Mustafa**]. In video games, GANs can be used as a method of up-scaling low-resolution 2D textures in old video games [**Wang**]. In medicine, GANs can be used to detect glaucomatous images, which an early diagnosis would help prevent partial or total loss of vision [**Bisneto**]. These are only a few examples of the capabilities that the GAN model provides.

CHAPTER III:
THE CLAS12 DETECTOR AND SPECTROMETER

Introduction

At the Thomas Jefferson National Accelerator Facility or Jefferson Lab, the CLAS12 detector (CEBAF Large Acceptance Spectrometer at 12 GeV) is part of the larger energy-doubling project of Jefferson Lab's Continuous Electron Beam Accelerator Facility. CLAS12 is situated in Hall B, one of the 4 experimental halls at Jefferson Lab. The primary purpose of the CLAS12 detector is used to study electro-induced nuclear and hadronic reactions (Fig. 10). The detector provides precise and efficient detection of charged and neutral particles, by utilizing complex experimental designs that minimize error [**Burkert**]. The

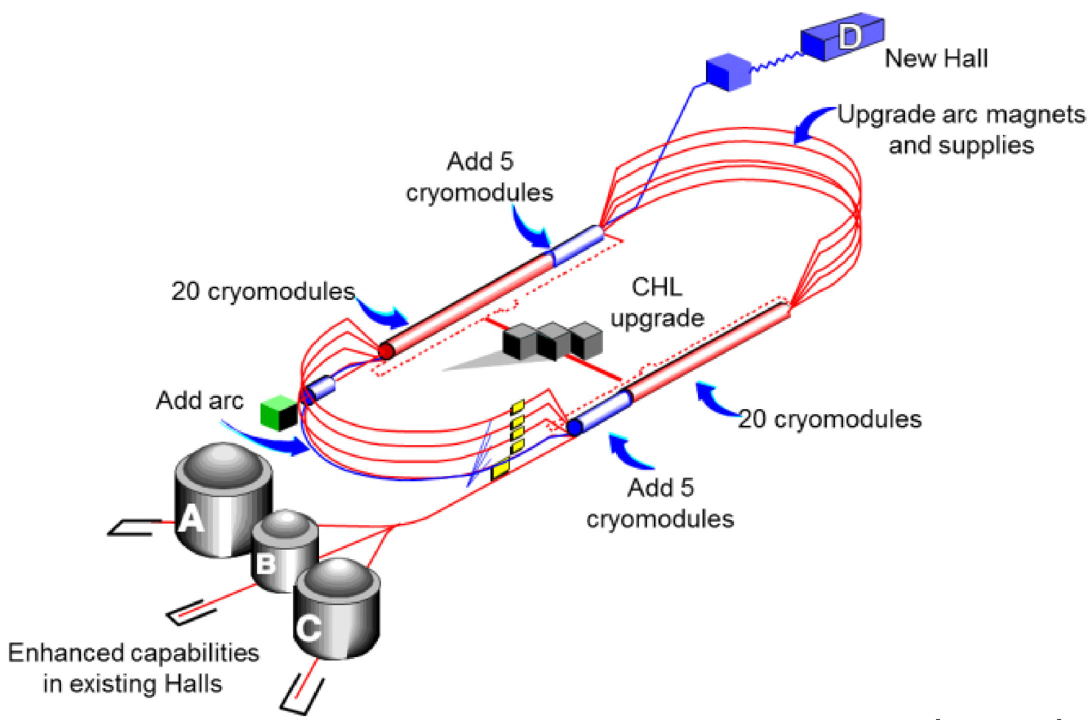


Figure 3: View of Jefferson Lab facility, hall and accelerator layout [**Burkert**]

CLAS12 spectrometer is designed around two large superconducting magnets. The first magnet is a torus magnet which provides a largely azimuthal field distribution that covers the forward polar angle range up to 35°. The second magnet is a solenoid magnet and the detector components that are contained within it cover from 35° to 125°, with full

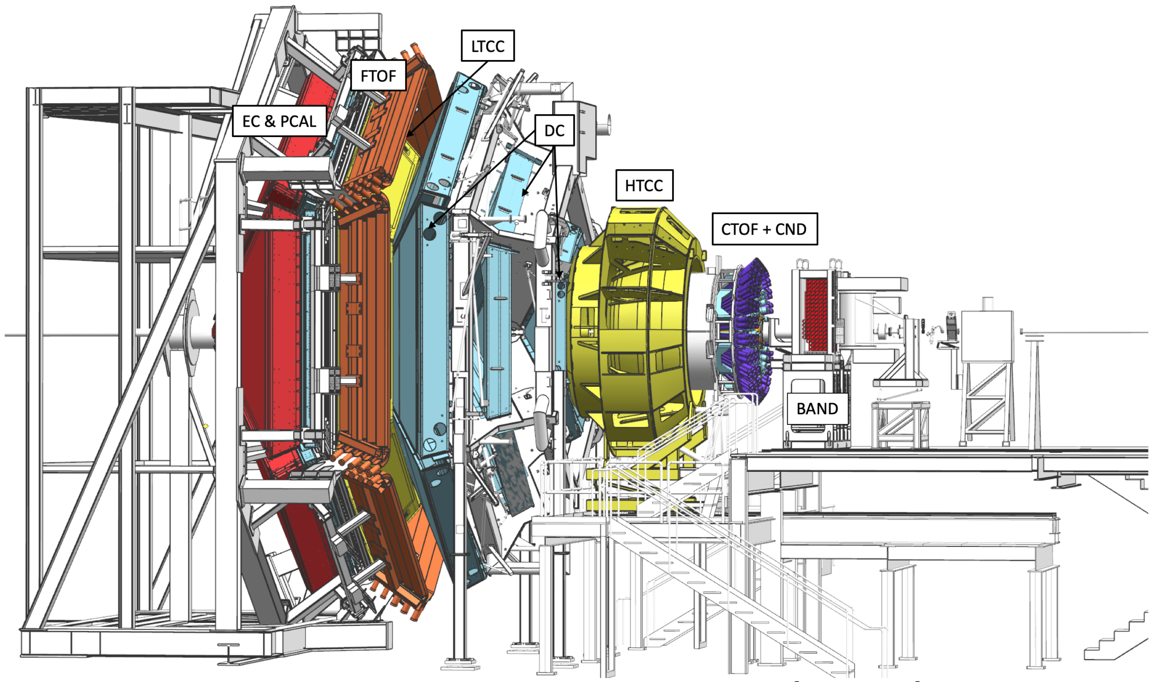


Figure 4: Side plot of the CLAS12 detector [**Burkert**]

azimuthal coverage. To reconstruct trajectory in the forward direction, drift chambers are used; moreover, in the central direction a vertex tracker results in momentum resolutions of <1%and <3%, respectively. Particle identification is handled by Cherenkov counters, time-of-flight scintillators, and electromagnetic calorimeters. The data is recorded by a data acquisition system (DAQ) working in conjunction with a fast triggering system, and allows for operation at a luminosity of $10^{35}cm^{-2}s^{-1}$ [**Burket**]. All of the CLAS12 detector capabilities are broadly being used to study the interactions and structure of nucleons, nuclei, and mesons, using polarized and unpolarized electron beams and targets for beam energies up to 11 GeV [**Burkert**].

CLAS12 forward electromagnetic calorimeter

Part of the CLAS12 detector, the forward electromagnetic calorimeter includes six independent lead-scintillator electromagnetic sampling calorimeters which are used to provide the primary electron trigger and extend photon and neutron detection capability to the CLAS12 system. In each calorimeter package there are two modules, the legacy Electromagnetic Calorimeter (EC) which was used from about 1996 to 2012 in the original CLAS detector, and the new pre-shower calorimeter (PCAL) which extends the total detector radiation

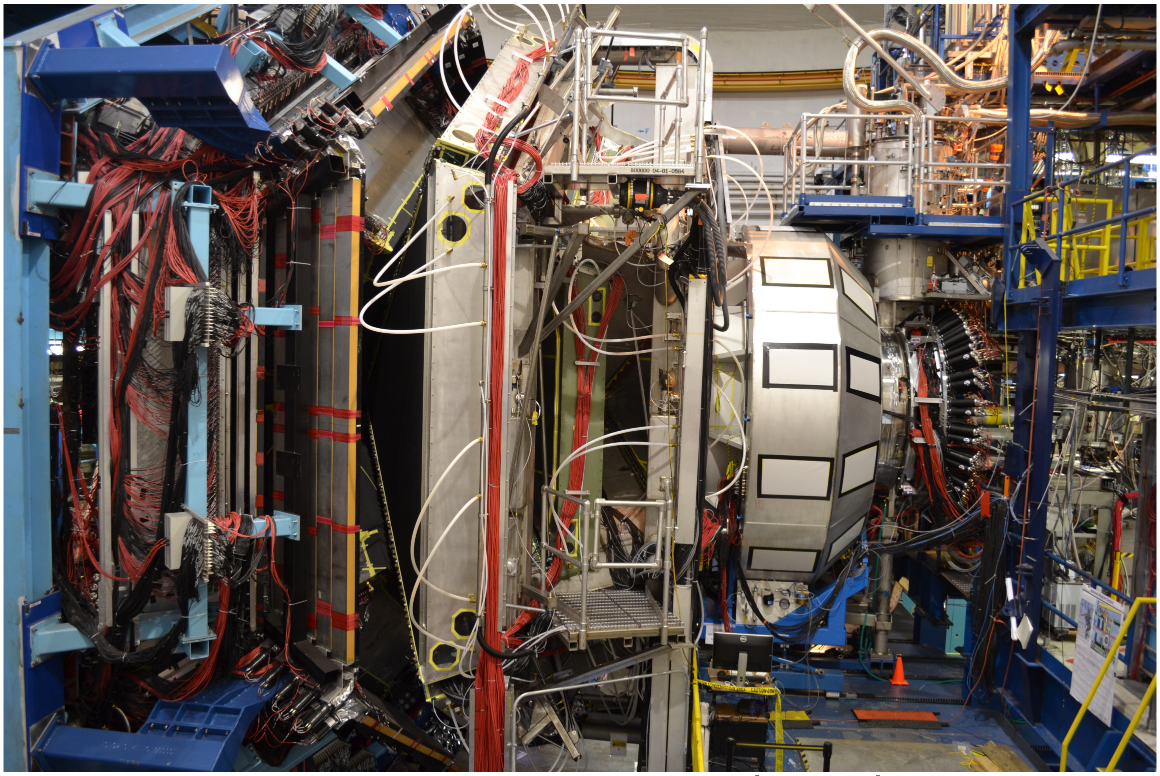


Figure 5: Photo of CLAS12 detector [**Burkert**]

length. To accommodate the higher momenta of the particles as the electron beam energy was increased from 6 GeV to 11 GeV. By extending this length, the calorimeter is able to fully absorb the electromagnetic showers induced by electrons and photons with energies up to 12 GeV [**Asryan**].

CLAS12 GEANT4 Simulation

After an experiment is ran through the CLAS12 detector, there is a large amount of raw data which is collected. To comprehend this data, a package called GEANT4 Monte-Carlo (GEMC) is used. The package simulates what is seen through the detector. This simulation is used to correct for detector inefficiencies [**Ungaro**]. The geometry is implemented through a database of GEANT4 volumes that were created using either the CLAS12 geometry service, the GEMC native API, or the CAD engineering model. The data is digitized with a plugin mechanism by routines specific to each detector and includes the use of the CLAS12 calibration database constants to produce both ADC and TDC response functions. There are theoretical models that produces the generated events and interface

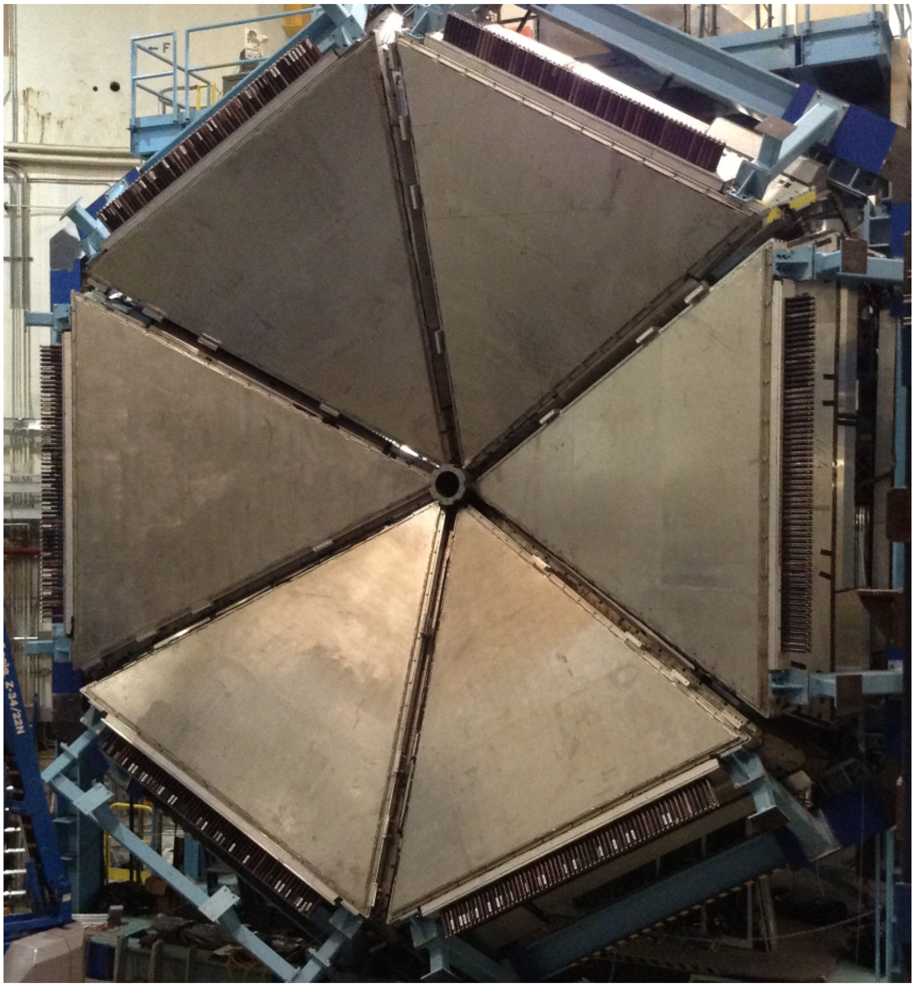


Figure 6: Photo of CLAS12 forward electromagnetic calorimeter and Preshower Calorimeter in 2014 during the installation process of CLAS12 [**Asryan**]

with GEMC through the LUND data format [**Ungaro**]. Finally, the merging of simulated data with real random trigger data provides a system that includes both beam and electronic background into the simulation of generated events to accurately model beam data from the CLAS12 detector. Comparing the experimental data to the theoretical data determines the simulation's performance. [**Ungaro**].

CLAS12 Software Framework

The CLAS12 experiment at Jefferson lab requires an accompanying software solution that can interpret the data in which the Spectrometer is gathering [**Ziegler**]. The data-stream processing framework in which was used is called CLARA, and it provides a service-oriented

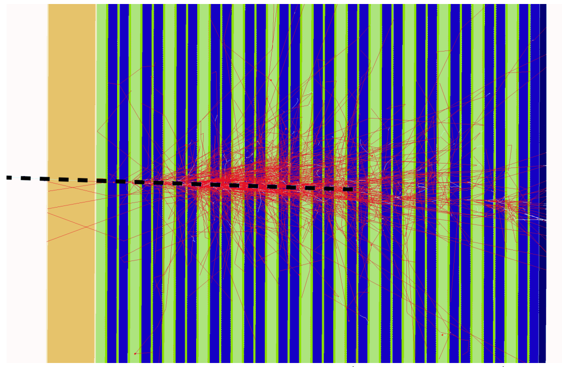


Figure 7: Example of 4 GeV electron track (dotted line) showering in the GEMC implementation of the ECAL geometry. [**Ungaro**]

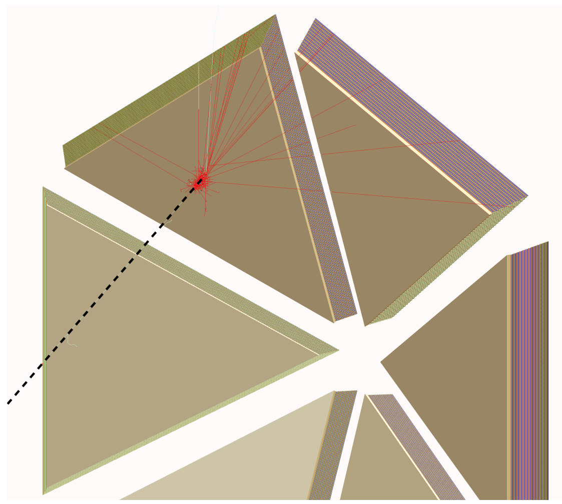


Figure 8: Example of particle shower inside of the calorimeter [**Ungaro**]

architecture in which to build the relevant software applications. GEMC will be used for simulation and CLARA is used to run the CLAS12 reconstruction software [**Ziegler**].

CHAPTER IV:
GANS NEURAL NETWORK AND THE CLAS12 ELECTROMAGNETIC CALORIMETER

Using GANs to Simulate

A deep learning model to enable high-fidelity fast simulation of particle showers in electromagnetic calorimeters can be applied to the CLAS12 system. The CLAS12 large acceptance spectrometer uses software framework, tools, and algorithms that support event reconstruction and analysis. It uses conventional methods that provide well understood algorithms; however, some of those algorithms in the data reconstruction process can be substituted with neural networks to substantially reduce data processing times [**Ziegler**]. Modeling particle showers in calorimeters is the most computationally demanding part of the simulation process. It can take minutes per event, even on cutting-edge platforms such as the scientific computing facilities at Jefferson Lab or on the Open Science Grid. The calorimeter simulation step for the CLAS12 detector takes up the vast majority of processing time. The elaborate structure of showers for particle identification along with energy and direction calibration make finding an approximate, fast simulation solution difficult. Simulating the electromagnetic calorimeter hits may use look-up tables for low energy interactions or rely on parameterized showers for fluctuations; that being said, these would be insufficient solutions for the CLAS12 system [**Paganini**]. If a GAN is used, an approximate, fast simulation solution can be provided that mitigates computational complexity.

Data

A detector simulation begins with a list of particles, and for each particle its type, energy, and direction is given. The type of particle determines when and how the particle interacts with the material (in this case lead) along its trajectory [**Paganini**]. The material interactions with the detector factorize; meaning, the energy deposited in a calorimeter by particles is the sum of the energy from each shower treated independently. An electromagnetic calorimeter is designed for the purpose of measuring the energy of leptons(primary electrons and positrons), as well as photons. [**Paganini**]. Essentially, the data which would be used in the generative adversarial network would be in the form of images, specifically, a 2D histogram. This histogram is converted into a matrix of values between 0 and 1, that effectively convert the image into a format easily readable by a computer. This matrix of values would be input to the generator, along with a random amount of noise. It is then the discriminators job to differentiate which photos follow a general set of rules determined by the patterns found in the data. The photos would consist of a three-dimensional particle energy signature that are divided into a series of three two-dimensional images[**Paganini**]. Particle identification and energy calibration in an electromagnetic calorimeter relies on transverse segmentation [**Paganini**]. This basically means that it needs to handle data that is divided into separate parts or sections, situated or extending across something.

Electromagnetic Shower

An electromagnetic shower is produced by a particle that interacts primarily or exclusively through the electromagnetic force, and is typically a photon or electron. It begins with a high energy electron, positron or photon that enters a material. At high energies, photons interact with matter using pair production. Pair production is when a conversion into an electron-positron pair occurs, interacting with an atomic nucleus or electron in order to conserve momentum. High-energy electrons and positrons primarily emit photons via a process called bremsstrahlung. The pair production and bremsstrahlung processes continue until photons fall below the pair production threshold, causing energy losses of electrons other than bremsstrahlung start to dominate [**Fields**].

CALOGAN Example

A simulated event is processed by GEMC, by taking a particle and traversing it through the detector and simulating the physical interactions that will occur in the different materials. This section follows an example called CALOGAN [**Paganini**]. The data collected can be broken down:

The figures demonstrate the electromagnetic calorimeter recording all of the particles paths after interacting with the material (lead). A particle is split into electrons and photons. As

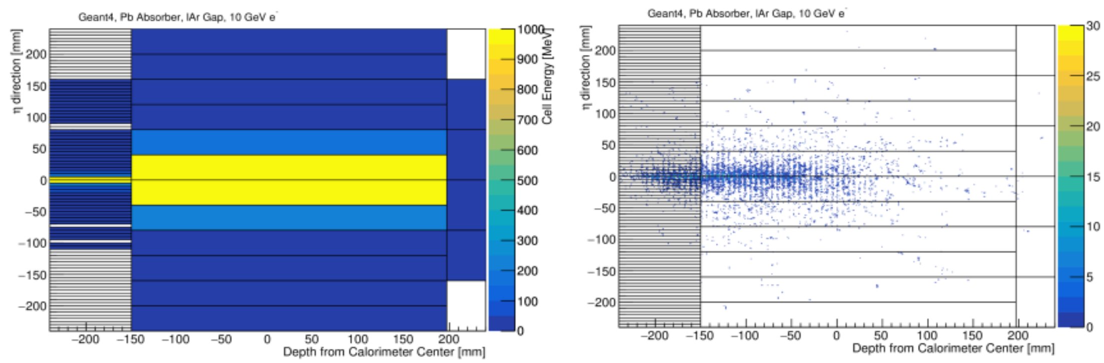


Figure 9: On the right: Each dot represents one energy deposit from GEANT4. The color of the dot encodes the energy. Most of the energy is lost in the absorber, and that can be seen in the absorber-gap structure. [**Paganini**]. On the left: A discretized version of the figure, in which energy depositions are assigned to individual, discrete detector cells. [**Paganini**]

the electromagnetic shower propagates through the calorimeter, it becomes wider and more difficult to measure the initial hit location. Each layer in the figure is a slice representing the particle distribution. There is a slice towards the beginning, one towards the middle, and one towards the end. The first layer has finer granularity for a more accurate position. It is highly segmented for better spatial resolution, followed by two less segmented layers that enable the entirety of the shower to be captured.

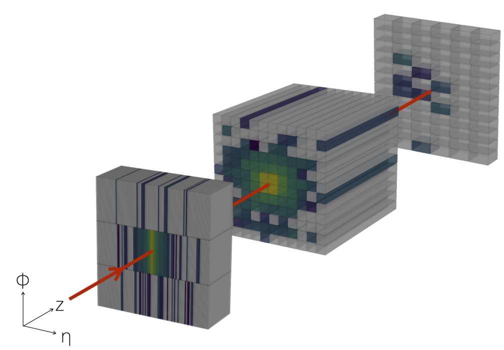
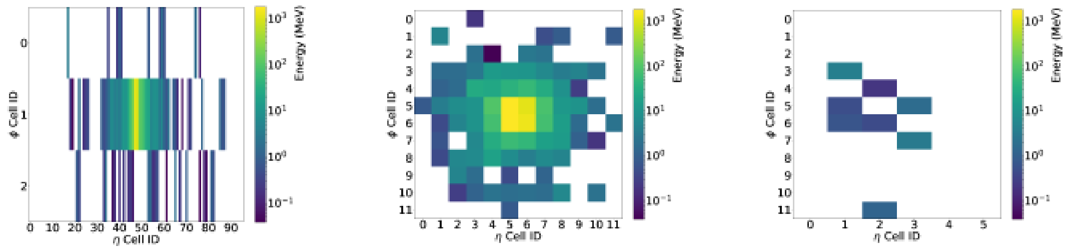


Figure 10: Three-dimensional representation of a 10 GeV e^+ incident perpendicular to the center of the detector. Not-to-scale separation among the longitudinal layers is added for visualization purposes.[**Paganini**]

The figures are 2D images that represent the 3D dimensional particle energy signature. The first layer can be represented as a 3 x 96 image, the middle layer as a 12 x 12 image, and the last layer as a 12 x 6 image [**Paganini**].



(a) First layer (b) Second layer (c) Third layer

Figure 11: Two-dimensional, per-layer representation of same shower as in Figure 4.2 [**Paganini**]

The detector simulation begins with a list of particles, and for each particle its type, energy, and direction is given. The way the particle interacts with the material (in this case lead) is dependent on the type of particle. The data would be in the form of photos. These photos would consist of a 3D particle energy signature that is divided into a series of three 2D images. Below is an example of the data in a histogram format.

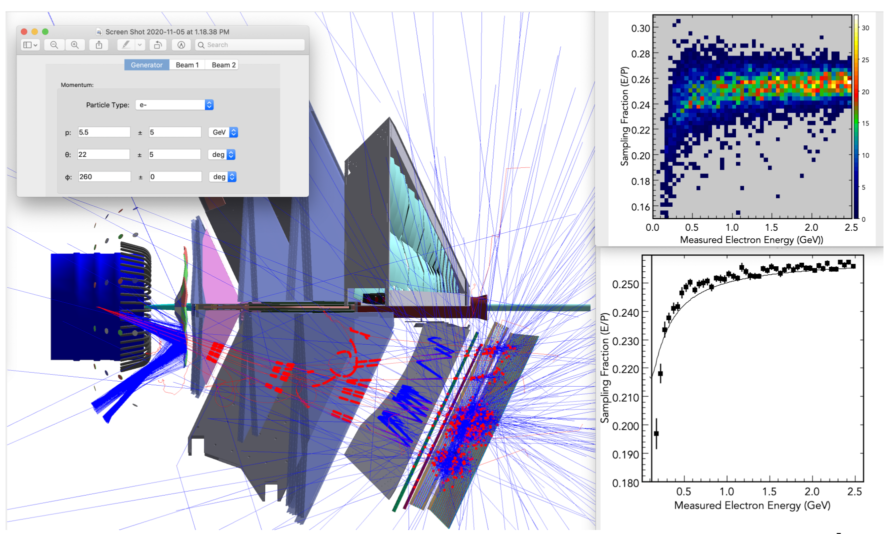


Figure 12: Model showing the sampling fraction as a function of energy [**Paganini**]

Why GANs?

Determining the best neural network for accomplishing the task depends on what is required and what has previously been accomplished. Convolutional neural networks handle visual image data well, and GANs deploys both a convolutional and a deconvolutional neural network. The combination of the two builds upon an ordinary convolutional neural network, and mitigates errors such as overfitting or oversimplifying. The deployment of GANs in an electromagnetic calorimeter is trivial when researching work previously accomplished with the model. GANs has been used to model electromagnetic showers in a longitudinally segmented calorimeter, and was able to achieve speedup factors comparable or better than existing full simulation techniques on CPU (100x-1000x) and even faster on GPU (up to 10^5x) [**Paganini**]. The promise GANs has shown with previous electromagnetic calorimeters, brings confidence that the neural network can be applied in a similar situation.

Tools Required

For the scope of this project, the plan is to use Python, Keras, Tensorflow, and NumPy. These tools would allow to create and train the GANs as needed, in a simplified format. Additionally, instead of using a personal machine for training and testing, jupyterhub would be used on the Jefferson Lab network. This would allow me to utilize the power of four titan RTX graphic cards, which speeds up the process.

Conclusion

For the CLAS12 electromagnetic spectrometer, a GAN can be used to significantly speed up the simulation process. Applying a GAN to a detector has been accomplished in one other case. If successful, this could potentially greatly improve the CLAS12 system's efficiency while still retaining accuracy. The scope of this project involves creating a generator and discriminator, then feeding it the particle shower hits captured by the electromagnetic calorimeter in the form of a 2D histogram. The objective is to make the GANs accurate enough to mimic the GEANT4 simulation at an equal level. If the accuracy is retained, the efficiency would enable simulations that would otherwise not be possible without an enormous amount of money and time. Eventually, the generator and discriminator can be implemented to GEMC, but for a one year project the focus is to create the GANs neural network and input the relevant data. Implementing GANs into GEMC would require an understanding of the C++ framework that was developed at Jefferson Lab and then modifying the neural network to fit into the framework. Eventually, the GANs created will be applied to GEMC and allow simulations that would not be possible due to a high cost that exceeds Jefferson Lab budget constraints.

CHAPTER V:
TIMELINE

Proposed Order of Completion