AI4EIC Hackathon

AI4EIC Hackathon Team

December 1, 2023

1 Introduction

1.1 Large Language Models

Since the release last year of ChatGPT by OpenAI, Large Language Models (LLMs) have gained a huge amount of attention and have been the topic of cutting-edge research across a number of domains. For example, LLMs have been used in the medical field for developing novel diagnostic procudures and individualized treatment plans, and aiding in novel drug discovery. The potential for LLMs on physics research is immense and largely untapped: by uncovering undiscovered patterns in complex data and through extensive, rapid literature reviews, and by aiding in code development. This year's AI4EIC workshop consisted of a number of talks focused on LLM applications for nuclear and particle physics [1][2].

1.2 GlueX BCAL

The GlueX Barrel Calorimeter (BCAL) is a 400 m-long, cylindrical electromagnetic calorimeter [3]. The BCAL can detect photon showers from 0.05 GeV to a few GeV with polar angle coverage of $11 - 126°$. The sampling calorimeter consists of layers of scintillating fibers interleaved with lead. As seen in Figure 1, the BCAL consists of 48 azimuthal regions, each of which is divided into four readout layers. Photons and neutrons that enter the BCAL leave an electromagnetic shower.

2 Details of the data

2.1 Simulation details

The training and testing data were generated by firing photons and neutrons from the Geant4 Particle Gun into a Geant4 model of the BCAL. These datasets use the Hall D recon-2019_11-ver01.2 reconstruction version and 4.35.0 simulation version. Particle samples are simulated in such a way to be approximately flat in reconstructed energy E = 200–2200 MeV and in z = 162–262 cm within the BCAL. This corresponds to a region of the phase space that is highly active within the calorimeter.

2.2 Data format

The Hackathon data is as follows:

- The data for each of the two parts is split into two CSV files: one for training and one for testing.

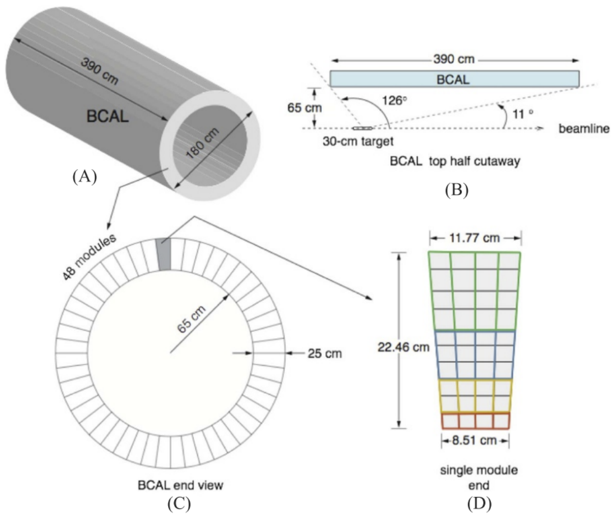

Figure 1: 2. Sketch of barrel calorimeter readout: (A) BCAL schematic; (B) a BCAL module side view; (C) and view of the BCAL showing all 48 modules and (D) an end view of a single module showing readout segmentation in four rings (inner to outer) and 16 summed readout zones demarcated by colors. Reprinted from [4].

- Each file consists of 14 variables which describe the particle showers in the BCAL (see Appendix A for the definitions of each of these variables).
- The training files contain an additional column for the event label: 0 for neutrons and 1 for photons.
- The testing files contain an additional column for the event ID.

2.3 Data location

Data files will be stored in the users' Amazon Web Service instances, under the directories */home/user/data/Part1* and */home/user/data/Part2*. The data files are named *AllTest_Part1.csv* and *AllTest_Part1.csv* (or *Part2*, for the Part 2 data).

3 Questions

3.1 Part 1

The first part will involve prompt tuning of ChatGPT-3.5 to generate Python code that will discriminate between photon and neutron events in the BCAL. All coding must be done by ChatGPT. The file you will submit to be evaluated must be a CSV file with two columns: Column 1 will be the eventID and Column 2 will be the PID (again, 0 for neutron and 1 for photon).

3.2 Part 2

The second part involves the same challenge as the first part, only the dataset is more difficult to classify.

4 Running Code and Submitting

After ChatGPT has output a Python program, a "Push" button will appear that will push the code to your AWS instance, under the directory *~/workspace*. After executing this code and generating your CSV file, you can submit the CSV file under the "Submit" tab of the web page by specifying the directory path and filename. Users may submit as many responses as they would like; the highest score from among all the team's users will be taken to be the team's final score. As stated previously, the submission file must have two columns, eventID and PID, in that order. The following is an example of the first couple rows of a submitted CSV file:

eventID	PID
549	1
127	0
⋮	⋮

5 Evaluation

The classified events in the submitted CSV files will be compared to the ground truth events from the simulation. The score for each question will be taken to be the fraction of events that are correctly classified, times 100, i.e. for N total events:

$$\text{score} = \frac{\sum_{i=1}^{N} \mathbb{1}_{index_{submit,i}=index_{true,i}}}{N} \times 100, \tag{1}$$

where $\mathbb{1}_{index_{submit,i}=index_{true,i}}$ is 1 if the i^{th} event index for the submitted file equals the true event index, and is 0 if they are not equal.

The overall score will be the sum of the scores for Question 1 and Question 2. Hence, the highest possible score for each question is 100, and the highest possible overall score is 200.

The uncertainty in the score will be calculated as follows:

$$\text{uncertainty} = \sqrt{\frac{1 - \sum_{i=1}^{N} \mathbb{1}_{index_{submit,i}=index_{true,i}}}{N}} \times 100. \tag{2}$$

Teams within uncertainty of each other will be considered tied. In that case, the tiebreaker will go to the most "prompt-efficient" team, i.e. the team that used the least amount of prompts.

6 Setting up the AWS Instance

Each hackathon participant will be given a unique AWS address. Use this address to ssh into the AWS instance with *ssh user@address* (replace "address" with the address we provide you; "user" is not a placeholder, literally type "user").

Participants will be responsible for setting up their own conda environments:

1. To set up a conda environment named *my_env*: *conda create -n my_env*
2. To install packages with conda: *conda install -c conda-forge package_to_install*
3. If the package does not exist in conda, use pip: *conda install pip; pip install package_to_install*

7 Hackathon Logistics

- Runs from 10:00 to 17:00 ET.
- Four rooms at CUA to work in: Caldwell Auditorium, Hannan Hall (Rooms 203 + 231), and Pryzbyla Center Room 321.
- Zoom link: https://jlab-org.zoomgov.com/j/1601365875 ?pwd=THNvaFV3T3podWJSenNYNlRUSzZKUT09 (Meeting ID: 160 136 5875, Passcode: 837159).
- Karthik Suresh, James Giroux and Patrick Moran will be available to answer questions in Caldwell Auditorium, in the AI4EIC Slack, or in the hackathon Zoom room.

Appendix A Data Feature Definitions

The overall shower features are denoted by a subscripted S. For example, S_z represents the shower z-position. T_z denotes the z-position of the center of the target and R denotes the inner radius of the BCAL. Most of the variables represent energy-weighted sums of the individual hits, represented with the summation subscript i.

- **r** $= \frac{1}{E_{tot}} \sum_i^N r_i E_i$ (i.e. the energy-weighted radial position, in cm)
- **LayerM_E** = The energy deposited in the Mth layer of the BCAL, in MeV ($M \in 1, 2, 3, 4$).
- **LayerMbySumLayers_E** = The fraction of the total energy deposited in Layer M of the BCAL ($M \in 1, 2, 3, 4$).
- **ZWidth** $= \sqrt{\frac{1}{E_{tot}} \sum_i^N E_i (\Delta z_i)^2}$, $\Delta z_i = (z_i + T_z) - S_z$ (i.e. the energy-weighted shower width in the z-direction, in cm).
- **RWidth** $= \sqrt{\frac{1}{E_{tot}} \sum_i^N E_i (\Delta r_i)^2}$, $\Delta r_i = R - r_i$ (i.e. the energy-weighted shower width in the radial direction, in cm).
- **TWidth** $= \sqrt{\frac{1}{E_{tot}} \sum_i^N E_i (\Delta t_i)^2}$, $\Delta t_i = t_i - S_t$ (i.e. the energy-weighted temporal width of the shower, in ns).
- **PhiWidth** $= \sqrt{\frac{1}{E_{tot}} \sum_i^N E_i (\Delta \theta_i)^2}$, $\Delta \theta_i = \theta_i - S_\theta$ (i.e. the energy-weighted shower width in the azimuthal direction, in radians).
- **ThetaWidth** $= \sqrt{\frac{1}{E_{tot}} \sum_i^N E_i (\Delta \phi_i)^2}$, $\Delta \phi_i = \phi_i - S_\phi$ (i.e. the energy-weighted shower width in the polar direction, in radians).

References

[1] A Large Language Model-based Assistant for the Electron Ion Collider: Towards a RAG-based Summarization, 2023. Talk at the 3^{rd} workshop on AI4EIC.

[2] An AI assistant for ATLAS. Talk at the 3^{rd} AI4EIC workshop, 2023.

[3] S Adhikari et al. The GlueX beamline and detector. *Nucl. Instrum. Meth. A*, 987(164807), 2021.

[4] C. Fanelli, J. Giroux, and Z. Papandreou. 'Flux + Mutability': a conditional generative approach to one-class classification and anomaly detection. *Mach. Learn.: Sci. Technol.*, 3(045012), 2022.