Hall D Computing

A Report for the 2021 Jefferson Lab Computing Review

The GlueX Collaboration
December 16, 2021

1. Introduction

The GlueX Experiment in Hall D has been designed to search for quark-anti-quark states or mesons in which the gluonic field binding the system contributes directly to the quantum numbers of the states. In order to do this very high statistics are needed leading to multi-petabyte sized data sets. In order to reliably extract physics from such data sets, the software and production of all data are managed centrally by the collaboration with the primary goal being to provide consistent reconstructed and simulated data in manageably sized data sets for physics analysis. Analyzers' interaction with the large data sets and Monte Carlo generation are principally through web-based interfaces, ensuring that all compute-intense activities are set up and run in a consistent fashion, leaving very little chance for error. This document gives an overview of all the processes and procedures that are involved in this production, as well as ongoing efforts to improve procedures through the use of AI and machine learning. It also projects future computing needs.

2. Hall D Online Skim System

The first phase of GlueX was successfully completed in 2018 where more than 3.5 PB was acquired with DAQ system data rates of about 400 MB/s. For GlueX Phase II, the data rate more than doubled to approximately 1.25 GB/s. This stressed the original system developed under Phase I which consisted of a single output stream written to a large capacity RAID disk server. While technically within specs for the individual components, the DAQ system exhibited instabilities when pushed to these higher rates. This motivated changes to the system to ensure stable high-intensity running. Specifically the raw data files would need to be distributed among several RAID servers in order to reduce the average rate any one server needed to support. Another issue that came up while processing the Phase I data was the considerable effort required to extract special calibration events from the stored data files. Calibration events were typically made from special triggers for things like LED flashers used by the calorimeters. The calibration events were mixed into the single output stream and were rare (less than 1%)

compared to physics events. The DAQ system implementation for GlueX could not be easily changed to write separate output streams for these events directly. Thus they needed to be extracted from the full raw data set starting from tape producing skim files. An ability to generate these skim files in the counting house before the raw data ever made it to tape would save considerable time and effort. Implementation was done using a separate, new system, the Hall-D Online Skim System (HOSS). Because HOSS needed to transfer a copy of the entire 1.25 GB/s data stream to a small compute farm in the counting house, it also became a natural way to distribute the raw data files among several RAID server partitions, reducing the I/O requirements for each partition. This is illustrated in Fig. 1.

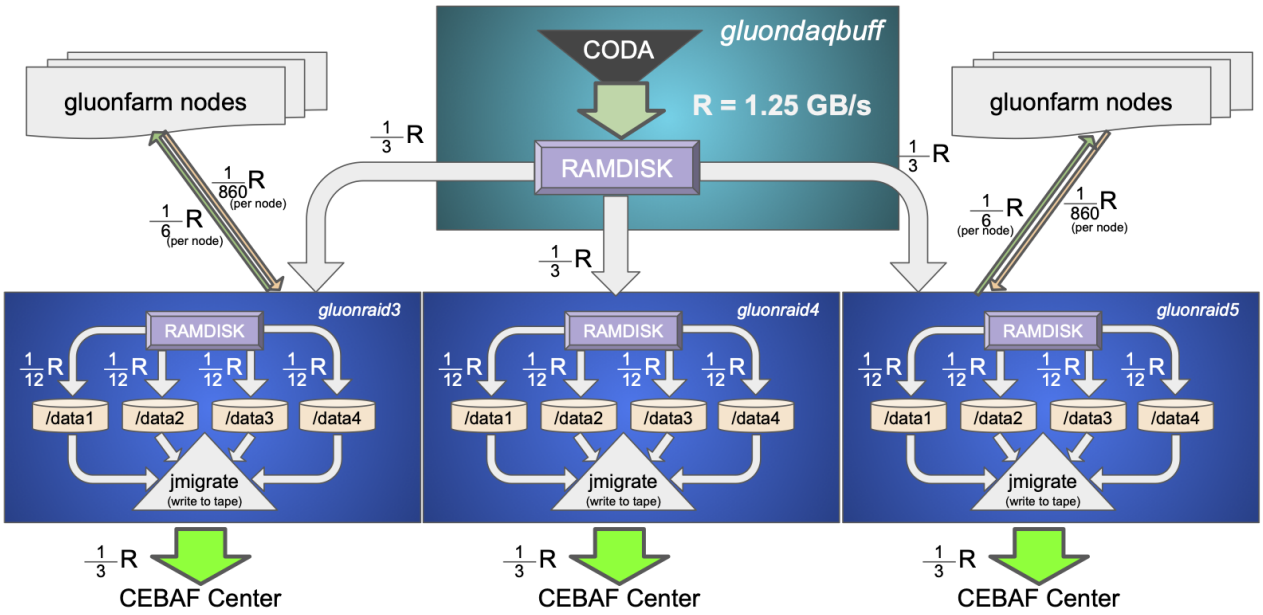

Figure 1: Illustration of how HOSS is configured for GlueX Phase II high-intensity running. *CODA* is the DAQ system that is configured to write data to a RAM disk. HOSS watches specific directories for files without open file descriptors and then moves them through the system.

The key orchestrator of HOSS is written in Python, but it relies on some key pieces of software to do the high-speed network transfers and CPU-intensive computations. RDMA is used over a 40-56 Gbps infiniband network fabric in the counting house. Custom RDMA servers written in C++ for Hall D are run as system services on almost all of the nodes in the GlueX online cluster. A custom tool was also developed that skims just the header of the 40-event blocks produced by the data acquisition to check if any of the events in the block is a calibration trigger that must be written out to a skim file. This avoids having to do computationally expensive dis-entanglement on every 40-event block. This savings in compute load allows HOSS to consume the entire data stream with a modest complement of 6 older compute nodes (Intel circa 2013). While scanning the headers for event type information, HOSS also records statistics for each data file: counts of each type of trigger as well as starting and ending event numbers. This information is written into a database on a MySQL server. It can be accessed either programmatically or via the web[1]. More details of the HOSS system can be found in Ref. 2.

3. Calibration

Prompt and efficient data calibration and validation is crucial in reducing the time between collecting the data and its availability for physics analysis. Over the past few years, there has been a focus on improving the stability of the readout firmware and in improving and automating calibration procedures, which has dramatically decreased the calibration time required. For example, in the running since October of this year the experiment to measure the η lifetime via the Primakoff effect (PrimEx-η) and the Short-Range-Correlation/Color-Transparency experiment (SRC/CT) have been able to reconstruct data for the analysis of important physics signals within hours of recording the data, which has proven important in evaluating the progress of these experimental programs.

The experience with running GlueX at high luminosity has shown several possible areas in which improvements may be helpful or required in upcoming years. The ability to perform calibrations using charged particles is very limited due to the CPU resources required for charged particle reconstruction, a limitation which has been made worse since the reconstruction time per event has increased 20-40% for the high-luminosity runs. The skims produced by the HOSS system described above have allowed us to decrease the amount of raw data processed offline for calibrations by a factor of three. Moving all non-tracking-based skims to this system would reduce our offline processing requirements by another factor of 3. The AI-based control system described in later sections promises to help reduce calibration time as well. These are all cases in which a small investment in increasing online processing capabilities will pay off in reducing the amount of offline resources required to calibrate the data and decrease the time to start physics analyses.

4. Reconstruction

As is often the case, reconstruction of the raw data is by far the most compute intensive activity that we perform. For the past few years we have performed reconstruction not only on the JLab farm, but also at several High-Performance Computing (HPC) sites. Successful applications for time have been made to NERSC and the Pittsburgh Supercomputing Center. Access to the Indiana University BigRed3 and BigRed200 supercomputers is a collaboration contribution and no application is needed. Table 1 gives statistics for each of these facilities.

At JLab the Scientific Computing group has developed a workflow manager SWIF that has been in operation for many years. It allows users to deal with job submission en masse, including efficient retrieval of input data from the Tape Library, with a built-in database keeping track of

individual job progress. Processing campaigns or “launches” have used a set of by-now-legacy Python scripts to manage submissions to the SWIF system (both for “Reconstruction” and “Analysis” launches).

For the HPC sites, job submission and management of input and output data was handled by a follow-on system, SWIF2. Initially SWIF2 extended the SWIF functionality to work at the HPC sites, but is about to be deployed at JLab as well. One major development needed for running on the HPCs was staging the 20 GB raw data files from tape, to local disk, and onto disk at the HPC site. Disk space at each stage needs management. Another wrinkle is that the details of job submission, disk space management, network transfer protocols varied at each facility, and these differences had to be managed on a site-by-site basis within the SWIF2 system. During the past year, the scheduling algorithm and our per event computing load changed for NERSC such that the workflow had to be modified from one input file per job to one run per job, where a run typically consists of 300 input files. The larger volume of computing on a per job basis gave vastly higher throughput under the new conditions.

Run period	**Length of processing**	**Fraction of events at each site**	**CPU used at each site in millions of core-hours**	**Number of jobs**
2017-01	1 Month	JLab (100%)	6	42165
2018-01	4 Months 1 Month	NERSC (81%) JLab (19%)	20.4 2.3	77603 16279
2018-08	2 Months 1 Month 1 Month	NERSC (52%) PSC (Bridges) (26%) JLab (22%)	9.25 0.81 2.1	24669 6990 13358
2019-11	4 Months 2 Months 2 Months 4 Months 4 Months	NERSC PSC (Bridges) PSC (Bridges2) BigRed3 JLab (57%)	12.4 2.33 4.03 3.56 23.9	45236 17752 19694 16392 119397

Table 1: Reconstruction processing.

5. Analysis

The full set of reconstructed data files (Reconstructed Event Summary Tape or REST files) is stored on tape and too large to be easily handled by individual analyzers. See Table 2. In order to reduce the size of the data sets accessed by analyzers, a central system was developed to process the REST data at JLab and extract reaction-specific ROOT trees.

Run period	# REST files	Size REST files [TB]	# Analysis Launches	# Channels	Σ Tree Size [TB]
Spring 2017	42k	117	51	1955	700
Spring 2018	84k	377	19	360	700
Fall 2018	48k	217	17	399	500
Spring 2020	210k	1,112	2	31	

Table 2: GlueX run periods and Analysis Launches

Users can request ROOT trees for reactions of interest via a web interface, shown in Fig. 2. Periodically, the submitted reactions are collected into a configuration file, which controls a workflow that produces all of the trees, an “Analysis Launch.” For each reaction, the GlueX analysis library inside the JANA framework creates possible particle combinations from the reconstructed particle tracks and showers saved in the REST files. Standard selection criteria are applied for exclusivity and particle identification before performing a kinematic fit, which imposes vertex and four-momentum constraints. Displaced vertices and inclusive reactions are also supported. Objects representing successful particle combinations (e.g. $\pi^0 \rightarrow \gamma\gamma$) and other objects are managed in memory pools, and can be reused by different channels to reduce the overall memory footprint of the process. With this scheme, up to one hundred different reactions can be combined into one analysis launch, processing the reconstructed data on multiple cores in parallel without large memory overhead.

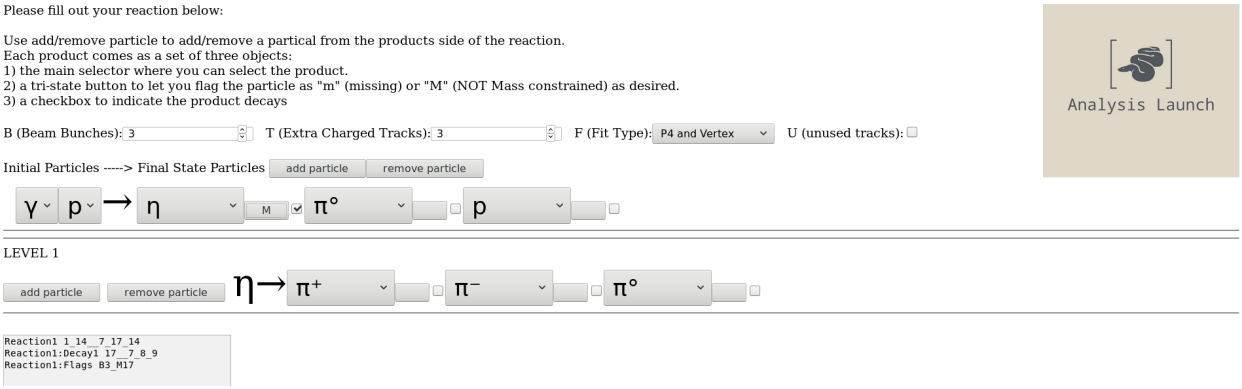

Figure 2: Web Interface for Analysis Launches.

If the kinematic fit converges for one combination of tracks and showers, the event is stored into a reaction-specific but generic ROOT tree. The size of the resulting ROOT trees strongly depends on the selected reaction. ROOT trees (about 200 per run) are merged into a single file whose size is suitable for copying a user's home institution for a physics analysis.
For a given run period, a new version of REST production or global changes to the selection criteria require that Analysis Launches be repeated for the new conditions. The total number of channels in Table 2 may therefore include multiple versions of the same reaction.

With nominal availability of JLab farm nodes, a typical analysis launch can be completed in one to two weeks. The elapsed time is limited by the latency due to retrieval of the REST data from the Tape Library. In the future, staging of files on SSD or on a distributed file system will help throughput.

6. Simulation

Simulations of the detector response are required in order to study the feasibility of measurements or apply corrections to data. The simulation of a typical reaction is split up into independent steps. The general flow consists of (1) event generation, (2) detector simulation where interaction of produced particles with detector elements is simulated, (3) smearing or addition of detector resolution and efficiency, (4) reconstruction, and (5) analysis data. The latter two steps are performed with the same reconstruction and analysis code as that used on real data.

6.1. Simulation Components

Event generation. A variety of event generators have been developed for different needs.

- **bggen** produces minimum-bias hadronic photoproduction events. It is based on a custom version of Pythia for high-energy photons and a compilation of known reactions for photon energies below 3 GeV.
- **genr8** produces events from a user-defined decay tree of hadronic resonances according to 2-body and 3-body phase space for a fixed photon beam energy.
- **genamp** is a collection of reaction-specific t-channel photoproduction generators. Samples are weighted by a user-selected set of partial-wave amplitudes.
- The **photon beam source** models the coherent bremsstrahlung process at the diamond radiator.
- The **beam conversion source** models pair+triplet production in the polarimeter target.
- The **Bethe-Heitler source** models e^+e^- and $\mu^+\mu^-$ pair conversion in various types of GlueX targets.

Detector Simulation. The original hdgeant simulation based on the CERNLIB GEANT3 library has been gradually superseded by hdgeant4 based on the Geant4 toolkit. Both programs utilize the same abstract geometry description and magnetic field maps, can read events from the same generators, and produce output events in the same format. The ability to directly compare the outputs from the two simulations has been very helpful throughout the transition period. The transition from primary reliance on hdgeant to hdgeant4 began in earnest in early 2018, and lasted for roughly 3 years.

Smearing. The mcsmear program reads in lists of raw hits from hdgeant4, and applies a set of transformations to them designed to imitate the detector response when passing particles deposit energy in sensitive elements. Run-dependent parameters describing the resolution functions and the efficiencies are stored in a database. Significant progress has been made in improving and calibrating these parameterizations over the past three years, especially the hit efficiency in the forward and central drift chambers as a function of drift distance. mcsmear also overlays accidental detector hits on top of the pattern of hits from hdgeant4. This is done by including hits from a set of random triggers obtained with each run. Simulated data is produced on a run-by-run basis so that the prevalence of accidental hits matches that of the real data.

6.2. Simulation with MCwrapper

To help users perform these five steps in an efficient and accurate manner, the tool MCwrapper[4] was developed to manage the entire chain. It is controlled with a configuration file

in which the user specifies software parameters such as package versions, or experimental parameters such as beam energy or polarization. MCwrapper can also query the run condition database to pull information that may vary on a run-by-run basis. MCwrapper can be invoked from the command line, but the recommended method is to use a web-based submission form. Fig. 3 shows a screenshot of the form. Here the user only has to make choices from dropdown menus and provide a path to a configuration file used by the event generator. This minimises the room for errors even further. Projects submitted via the webform are first checked against a database to make users aware of other projects using the same configurations to avoid duplication of effort. Projects undergo automated small-scale testing and upon passing, jobs are submitted to the Open Science Grid (OSG). As of November 19, 64 unique users have submitted 1,874 projects, which ran more than 2.3 million jobs and produced about 40 billion events. The total CPU time used for successful projects is about 10 mega-core-hours.

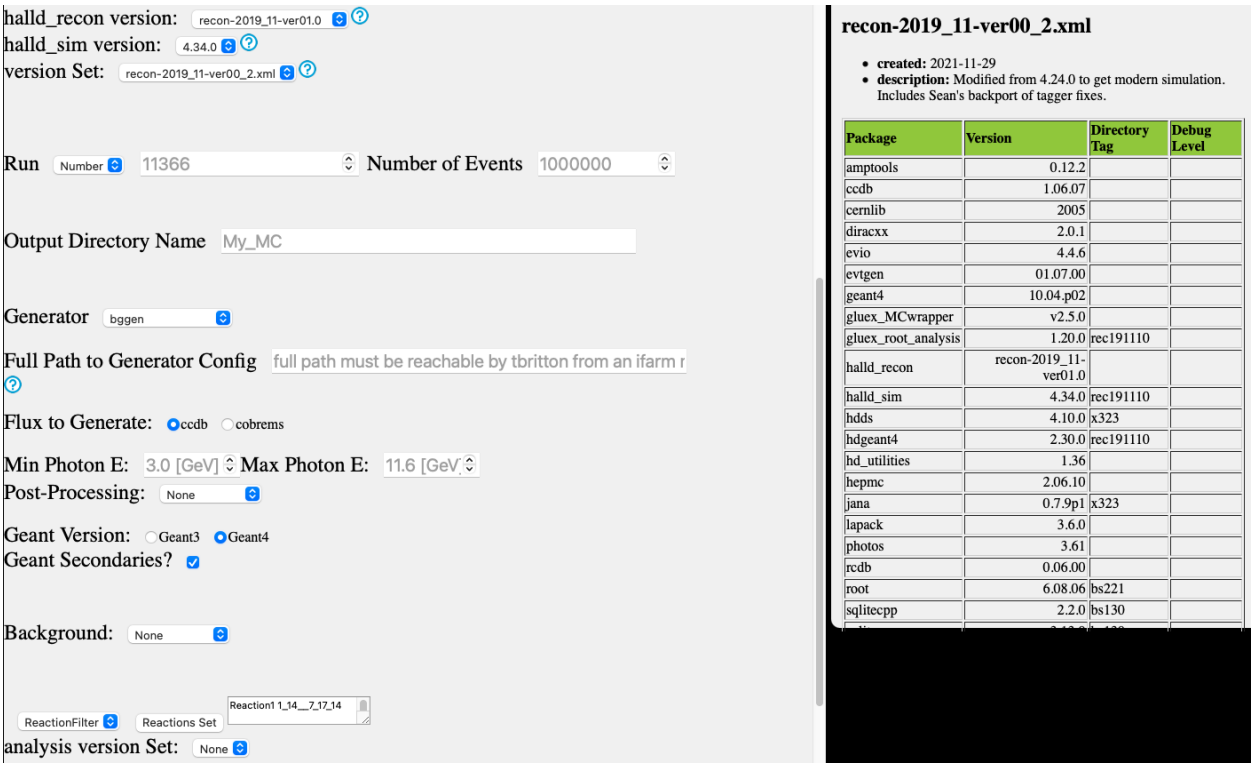

Figure 3: Screenshot of MCwrapper submission web interface.

7. Artificial Intelligence and Machine Learning

Hall D and its allies have pursued several projects in the area of artificial intelligence (AI) and machine learning (ML).

7.1. FCAL shower Classification

For the GlueX forward calorimeter (FCAL) a multi-layer perceptron algorithm was implemented to differentiate between low energy photons and split-offs occurring from hadronic interactions in the detector. The algorithm takes eight inputs about the showers mainly focused on the energy distribution and geometric shape of the particle shower in the detector. It returns a quality score between zero and one with a score of one corresponding to a true photon. Studies were conducted using several metrics including figure of merit (FOM) studies, where FOM = $\sqrt{S}$ / (S+B). Here S denotes the signal yield and B denotes the background yield. The optimal quality requirement for the figure of merit yields a background reduction of 60% and a signal retention of 85% on inclusive neutral pion data. Fig. 4 shows two-photon invariant mass distributions with and without a cut on the quality score.

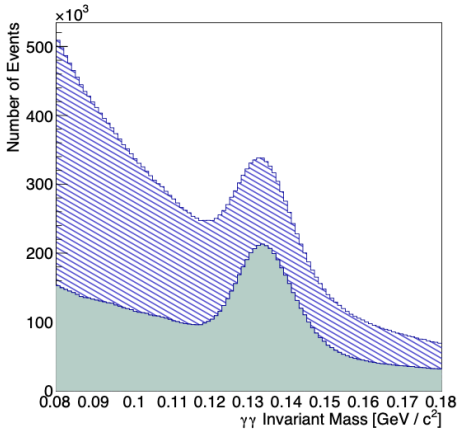

Figure 4: Reduction of combinatoric background resulting from a quality requirement of 0.5. Data with no quality requirement is shown in striped blue, and the data with the quality requirement is solid green.

7.2. Experimental controls predicting drift chamber constants from EPICS data

For the GlueX Central Drift Chamber (CDC) calibration involves many iterations over a subset of collected data on a per run basis. The DOE funded proposal “AI Calibration and Control” seeks to design, develop, and deploy AI systems which can both calibrate and control the detector.

There are two major parts to the calibration of the CDC: the gain correction factor (GCF) and the time-to-distance calibration (TtoD). Much of the work performed thus far in the 3 year project has been focused on the GCF. Using historical data on the environmental conditions (*e.g.*,

atmospheric pressure, CDC high-voltage-board currents, *etc.*) AI models were developed which, to varying degrees, accurately produced the GCF as determined with traditional methods, as shown in Fig. 5.

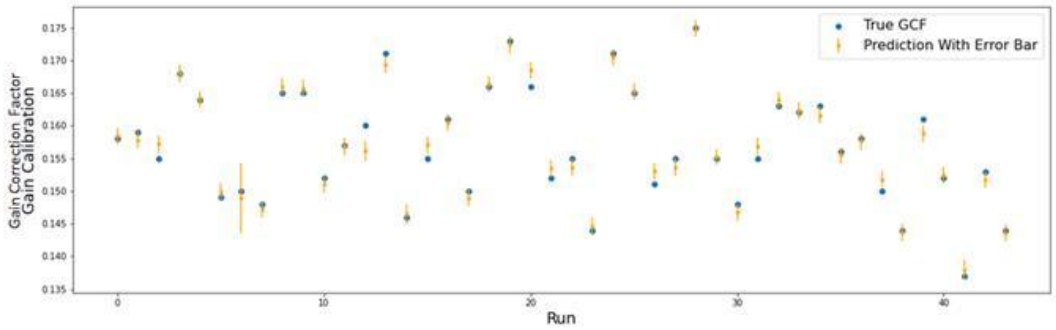

Figure 5: An example of the GCF across test runs and the AI model's prediction with uncertainties.

Additionally, during the PrimEx-η running a system was deployed which would recommend high voltage (HV) settings based on the prior minute of environmental conditions. The goal of HV control is to stabilize the gains, ideally around 1.41, potentially removing the need for gain calibration completely. The system was able to stabilize the gains more so than running using a nominal setting, as shown in Fig. 6.

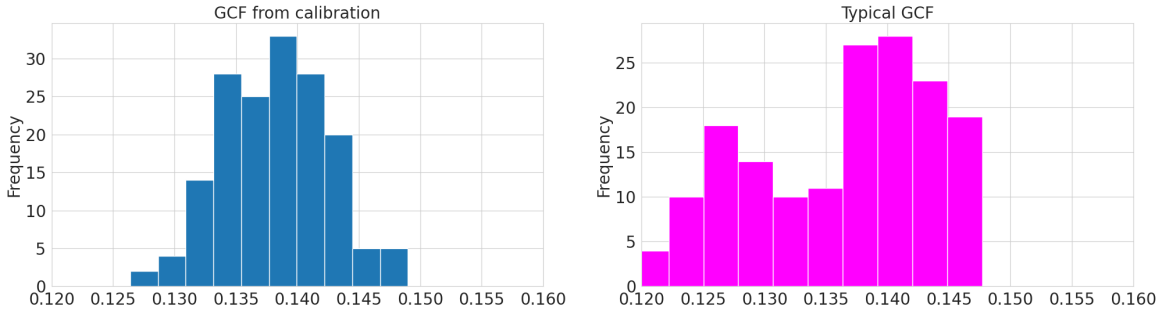

Figure 6: Left: the distribution of GCF found after AI HV control. Right: the typical gains using only nominal voltages.

7.3. FPGAs for Online Processing

The growing computational power of modern FPGA boards allows us to add more sophisticated algorithms for real-time data processing. Some tasks, such as clustering and particle identification, could be solved using modern machine learning algorithms which are naturally suited for FPGA architectures. To demonstrate the operating principle of the ML FPGA, and estimate the performance of the filter, studies have been done with data from a test beam setup with a GEM transition radiation detector and a E&M calorimeter prototype. Details can be found in Ref. 3.

7.4. Hydra

Hydra[5] is an AI-based system in active deployment in Hall D. It is, at its core, an extensible framework for training and managing AI models for near real-time data quality monitoring. At present Hydra manages a little over 12 models (based on Google's Inception v3 topology), each of which is responsible for monitoring a single plot (in image form) from one sub-detector. These models categorize images into broadly labeled buckets (e.g. "Good", "Bad") focused on go/no-go issues. The results of analysis by Hydra are collected and displayed, with their corresponding image, on a globally accessible webpage. See Fig. 7 for an example screenshot.

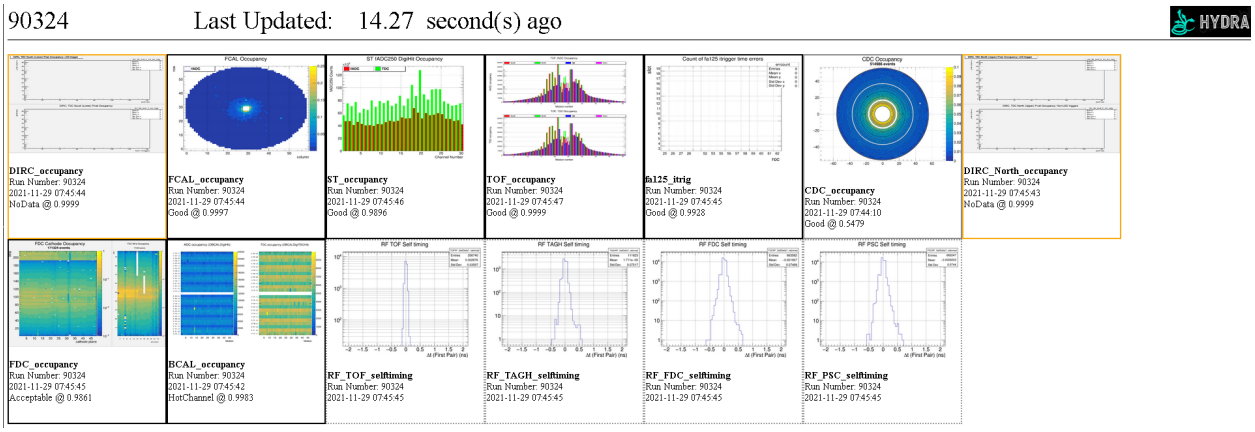

Figure 7: An example screenshot from the live webpage.

So far Hydra has detected problems that have gone unnoticed by detector experts and shift crews and can be expanded to include a more specific diagnosis to problems. The system itself is integrated with Hall D's RootSpy system through the RootSpy-AI (RSAI) process, which is responsible for converting the ROOT histogram files to images. From there a process intelligently re-scales the over-sized images into a form acceptable for feeding into the appropriate model; inference is performed by the trained model and the results are published via an inter-process message (using ZeroMQ). A "Keeper" task subscribes to these messages and processes the results. In all cases a non-biased sample of images are collected at a configurable rate, per image type, to be used for spot checking model performance and as an aid in future retraining. Additionally, all images that Hydra deems "Bad" are collected for future retraining.

7.5. Charged-pion polarizability experiment and Bethe-Heitler pair identification

The charged-pion polarizability experiment in Hall D is a precision measurement of low-energy QCD using pions pairs produced via the Primakoff effect. Making such a measurement relies on the ability to distinguish pions produced at threshold from Bethe-Heitler muon and electron pairs.

Separating muons from pions is significantly more difficult than separating pions from electrons. The primary difficulty is that muon, like pions, generally do not produce electromagnetic showers forward calorimeter (FCAL). Construction of six multi-wire proportional chambers (MWPCs), to be deployed between several layers of steel absorbers, has been completed to assist in π/μ separation, and a neural net that combines information from the forward drift chambers, calorimeters, and the MWPCs is under development.

The simpler problem of identifying Bethe-Heitler electron pairs has been studied with already-taken GlueX data. A multi-layer perceptron neural net was trained using ROOT's TMVA package using three features from the forward calorimeter and drift chambers–the ratio of FCAL energy to charged track momentum, the E9E25 shower ratio (the ratio of energy deposited in a 3x3 block grid to that for a 5x5 block grid), and the distance between the charged track projection and the centroid of the calorimeter shower. The electrons were trained on simulated data and pions on real data from the ρ^0 peak. Fig. 7 shows results from the study.

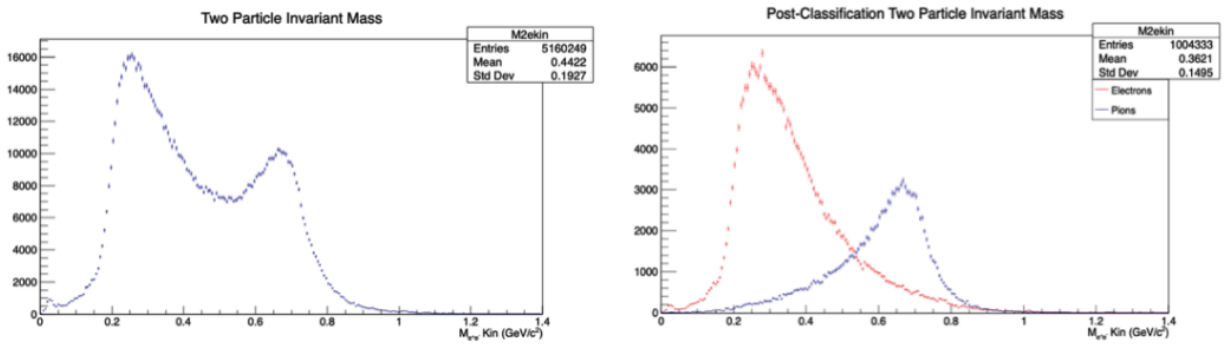

Figure 8: Left: input GlueX data containing both electrons and pions prior to classification. Right: the neural net applied to the input data twice, first selecting for pions (blue), then electrons (red).

7.6. Particle Identification with AI

The charged particle identification (PID) at GlueX uses both a traditional and a machine-learning-based approach. The latter employs an autoencoder neural network which consists of an encoding and a decoding stage. The training of such a neural network does not require any labeled data. Once trained, the encoding stage allows compression of the data from a N-dimensional feature space to a n < N-dimensional latent space. The compressed data is projected back to the original feature space by the decoding stage. Any difference between the data that is fed into the autoencoder and the data used for training will result in a variance after the encoding stage. This variance will be decoded as an anomaly and can be used to discriminate between different particle tracks.

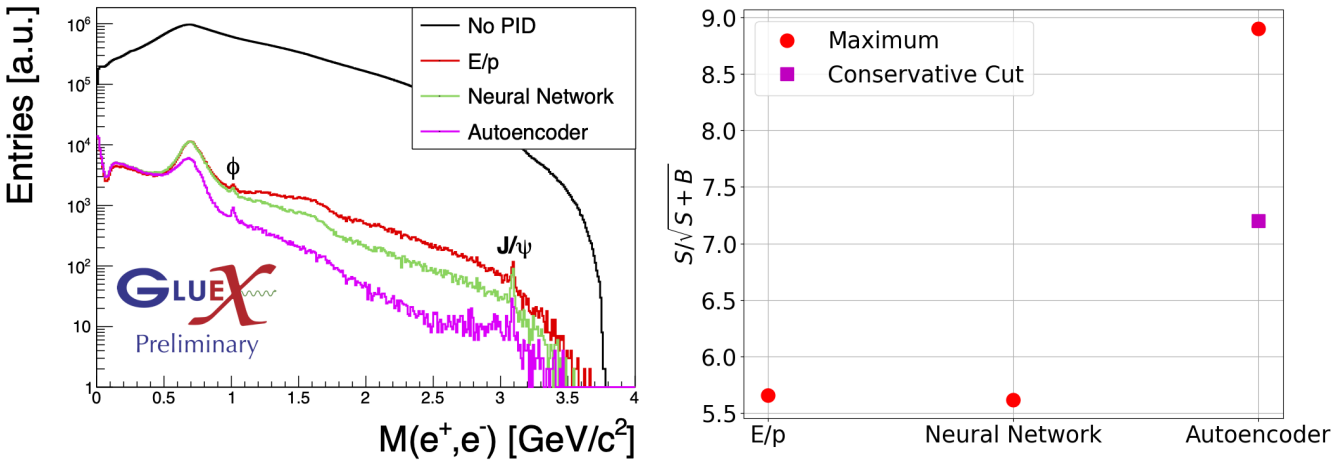

Figure 9: Left: Dilepton invariant mass deduced from a fraction of the GlueX spring 2018 data, after applying different PID methods. **Right:** Identification significance for J/ψ → e^+e^- events, using the PID approaches discussed in the text below.

Fig. 9 summarizes the lepton identification performance for J/ψ → e^+e^- events. Three PID methods are compared: (i) E/p cut: A cut on the energy deposit in the GlueX calorimeter, divided by the momentum, (ii) a neural network, trained on simulated data, containing leptons and pions, and (iii) an ensemble of autoencoder networks, one for each particle (lepton or pion) and charge. The left panel of Fig. 9 shows the dilepton invariant mass after the various algorithms have been applied. The autoencoder approach gives the most effective suppression of ρ→$\pi^+\pi$ events (the major background in this data set) and helps to reconstruct J/ψ→e^+e^- events, as well as φ→e^+e^-. A significance scan with respect to detecting J/ψ → e^+e^- was performed for each approach (see the right panel of Fig. 9). The maximum significance is obtained for the autoencoder ensemble (compare red dots in the right panel of Fig. 9). Even using a conservative cut on the ensemble probability shows a comparably higher significance.

8. GPU resources for amplitude analysis

The final step in the GlueX analysis chain often involves a single analyst performing an amplitude analysis on a data set. Amplitude analysis involves an unbinned multi-dimensional likelihood fit to the data set and has, for decades, been the standard technique for extracting resonance properties from data. Very roughly the computing cost of a fit is given by the product of the number of events being fit and the complexity of the model. The large GlueX data set and sophisticated phenomenological models developed by the Joint Physics Analysis Center (JPAC) drive both terms in this product. The problem is ideal for parallel computing on GPUs, and the collaboration is currently using the AmpTools library, which initially supported NVIDIA

GPU-accelerated fitting about ten years ago and has undergone many iterations of improvement and optimization in the past decade. While AmpTools has methods to optimize memory use and also distribute a single fit across multiple GPUs (even on different nodes via MPI), the limitation one often runs into is memory. If all of the data needed to perform the unbinned fit can't be loaded into GPU memory, then GPU acceleration is not a viable option. The new NVIDIA A100 and V100 GPUs, which are also effectively deployed for machine learning applications, provide up to an order of magnitude more memory than previous generations of GPU and are ideal for using computationally complex models to fit the large GlueX data sets.

Our experience is that it is relatively easy for a single analyst to saturate the available GPU resources on the GPU enabled nodes on the JLab SciComp cluster (3 with TitanRTX cards and 3 with Tesla T4 cards = 44 GPU cards in total). For a typical analysis, a standard workflow requires multi-dimensional binning resulting in ~100 independent fits, each running for several hours on a single GPU to fit a given model. With hundreds of possible models to fit and many analyses being performed in parallel the existing resources will soon be oversubscribed, given that their usage for machine learning applications are also growing rapidly. In addition the cost of the high-memory GPUs that are ideal for amplitude analysis prohibits many institutions from making an investment in this hardware. The collaboration would benefit from an enhanced pool of state-of-the-art high-memory NVIDIA GPUs that could be shared with other activities at the lab that can exploit this computing architecture.

9. Reconstruction on the Open Science Grid

A demonstration system has been developed, deployed and tested to do GlueX event reconstruction on the OSG. Each 20 GB raw data file is split into 60 to 70 small files and a single OSG reconstruction job is run against each one. This allows us to run single-threaded jobs taking 2 to 3 hours, opening access to opportunistic resources. Results are copied back to a local host and merged to produce one output file per input raw data file. A PostgreSQL database is used to keep track of all the partial files. We hope to roll out the system, at scale, in the coming year.

10. Areas for Improvement

There are several areas in which we would like to do better.

- **Data Catalog**. Our workflows, in total, produce millions of files. We would benefit from a global data catalog not only to keep track of what files we have and where they are, but also what files that we expect to be produced have in fact *been* produced. Another desirable feature would be to validate files, according to some user-defined criteria, as they are produced and record results of the validation. Many of the workflow managers that we use have databases underpinning their work, but those are not instrumented for

direct user access to facilitate custom queries and are generally aimed at tracking jobs and not files. We are particularly interested in pursuing solutions that leverage work by other collaborations/labs and adapting them to our needs.

- **Work Flow Management**. We need a mechanism to couple work flow management systems more tightly to any future data catalog. Lack of coupling defeats many of the advantages of a data catalog.
- **Continuous Integration (CI)**. We have a system for CI but the tests are limited in scope.
- **Comprehensive Testing**. Global testing of reconstruction and simulation is done, but there is not a good way to track changes in performance over time.
- **Unit Testing**. We do very little unit testing and have not developed a paradigm for implementation.
- **Documentation**. We have recently focused some resources on this area. New collaborators have complained that documentation is hard to find and often out of date, among other age-old problems. We have recently formed a documentation task force to take a comprehensive look at how we can improve in this area.

11. Conclusions

It has been a busy period for GlueX since the last Computing Review three years ago. All stages of the scientific enterprise from data taking, through data analysis, and publication of results are now in full flight. Many lessons have been learned and areas of improvement have been identified. There are also many ideas for future developments, some of which have been pioneered by other experiments, others that are more speculative. All of these endeavors would benefit from more human resources deployed at the interface of physics data analysis and software engineering.

References

1. HOSS webpage: https://halldweb.jlab.org/hoss
2. HOSS paper: https://doi.org/10.1051/epjconf/202125104005
3. FCAL Shower Classification ML paper: https://arxiv.org/pdf/2002.09530.pdf
4. FPGA ML paper: https://misportal.jlab.org/ul/publications/view_pub.cfm?pub_id=16832
5. MCwrapper paper: https://doi.org/10.1051/epjconf/202024503028
6. Hydra paper: https://doi.org/10.1051/epjconf/202125104010