Artificial Intelligence in CLAS12
Artificial Intelligence/Machine Learning for Physics Applications

G. Gavalian (Jefferson Lab)

Angelos Angelopoulos (CRTC)
Polykarpas Thomadakis (CRTC),
Nikos Chrisochoides (CRTC)
Department of Computer Science,
Old Dominion University, Norfolk, VA, 23529

Norfolk (May 9, 2023)
Outlook

Charged Particle Tracking:

- Track identification in Drift Chambers
- Drift Chamber Data De-Noising
- Impact on the experiment outcome
CEBAF
- 12 GeV electron beam distributed to 4 experimental hall
- Each experimental hall contains a detector system for specific experiments

CLAS12
- CEBAF Large Acceptance Spectrometer (CLAS12) Located in Hall-B
CLAS12 Detector

- 2 super layers in each region
- 6 wire planes in each super layer with 6-degree tilt relative to each other, (112 wires in each plane)
- Clusters in each super layer are considered part of the track trajectory

- Charged particle tracking is computationally extensive (about 80% of data processing time)
- The multi-particle final states produce numerous clusters in each sector which have to be analyzed to find the right combinations of clusters that form a track
- Identifying correct cluster combinations can speed up the tracking process and improve efficiency

![CLAS12 Detector Diagram]
Physics Results

- True tracks are identified by conventional algorithms from real data.
- One negative and one positive track (different curvature due to magnetic field)
- False tracks are constructed by interchanging randomly one or two clusters with the clusters from the other track in the event

- The average wire position in each super layer is used as an input to Multi-Layer Perceptron (MLP)
- The network is trained on 6 inputs and produces three outputs:
  - False track
  - Negative Track
  - Positive Track
Corruption Auto-Encoder

- An auto-encoder is composed of an encoder and a decoder sub-models. The encoder compresses the input and the decoder attempts to recreate the input from the compressed version provided by the encoder.
- Typically used for de-noising, but can be used for fixing glitches (our case).

The network predicts the missing cluster position with a precision of 0.36 Wire.

Use Auto-Encoders to fix the missing cluster (provide a position).

Good reconstructed tracks are used to generate training samples by removing one cluster from each super layer.
Putting things together

Classifier picks the correct track from 6 super-layer combinations

Remove all clusters belonging to identified track

Construct pseudo-clusters for all 5 super layer combinations using Corruption Auto-Encoder

Identify tracks using 6 super layer candidates with pseudo-clusters

Voila!
Physics Results

AI-assisted track candidate classification and Inefficiency Reduction Auto-Encoder

\[ e^p \rightarrow e'^\pi^+(X) \]

\[ e^p \rightarrow e'^\pi^+\pi^-(X) \]

- Single particle efficiency increases by \(\sim 10\%\).
- The impact on physics for a multi-particle final state is dramatic (20\% for the two-particle final state and \(\sim 35\%\) for the three-particle final state).
- The tracking code speedup is \(\sim 30\%\).

\[
\frac{f=1.00+0.0040x}{f=1.00+0.0022x}
\]

\[
\frac{f=0.99+0.0024x}{f=1.00+0.0040x}\]

\[
f=1.00+0.0022x\]

\[
f=1.00+0.0040x\]
~35% gain in physics
Moving to higher Luminosities
De-Noising

Performance of track identification for higher luminosity

- Pythia simulated physics reaction:
  \[ ep \rightarrow e' \pi^+ \pi^- p \]

- Data for each luminosity (beam current) is created by standard background merging software.

- For each luminosity the yield of missing protons is calculated in:
  \[ ep \rightarrow e' \pi^+ \pi^- X \]

- With increased luminosity the efficiency of reconstructed three particle final state drops sharply.

- Even with the power of AI-assisted tracking (capable of resolving the combinatorics) the efficiency drop follows the same trend.
De-Noising

- In high luminosities the noise level increases and forming clusters (or segments in each chamber becomes challenging)
- This results in loss of clusters and AI-assisted tracking can no longer help with combinatorics resolution

CLAS12 Event Display Examples (Drift Chambers)
De-Noising

- Convolutional Auto-Encoder is used to de-noise raw data from drift chambers.
- The network is trained on reconstructed data with track hits isolated from raw DC hits.
- The network is able to isolate hits that potentially belong to a valid track through drift chambers.

Network Performance Summary

RAW DATA | GROUND TRUTH | De-NOISED

Beam Current (nA)

Efficiency

- Hit reconstruction efficiency
- Noise reduction efficiency

45nA | 50nA | 55nA | 90nA | 100nA | 110nA
The reconstruction is run on simulated data with a merging background for different incident beam currents (luminosity).

The simulated three-particle final state is analyzed to measure yield for de-noised data and for conventional tracking.

At standard running luminosity, the de-noising slightly increases the yield compared to AI-assisted tracking.

With increased luminosity, the de-noising helps to increase the yield significantly compared to conventional and AI-assisted tracking.

Simulation underestimates the gain in yield significantly. In data the gain is much larger.
Current Workflow

- CLAS12 Reconstruction software is based on SOA (CLARA) approach, where each detector reconstruction runs as a separate service.
- The data reconstruction workflow now includes de-nosier running prior to standard clustering and AI-Assisted tracking running prior to DC track finding.
- Drift Chambers code runs tracks suggested by AI-assisted tracking through Kaman-filter for final track parameter calculations.

Running at standard conditions (45 nA beam current) the AI increased the yield of missing protons by 51%.

The improvement in yield is reaction and kinematics dependent, and for some event topologies reaches even 83% (J/psi with 3 particles detected final state).
Summary

- CLAS12 uses three neural networks for track reconstruction in forward drift chambers:
  - De-Nonoiser: Convolutional Auto Encoder Network
  - Corruption Recovery Network: Multi-Layer Perceptron AutoEncoder
  - Track Classifier: Multi-Layer Perceptron Neural Network
- The combined effect of three neural networks resulted in increase of single particle efficiency $\sim 15\%-18\%$.
- The resulting increase in statistics for physics observables is $\sim 50\%-80\%$.
- Implementation of AI track identification also resulted in tracking code speedup of $\sim 35\%$.
- The use of neural networks in track reconstruction pave the way for high luminosity running where conventional methods can not be used.
- Future: working on neural networks for other detectors
Backup Slides
De-nosing Performance Multi-Threaded

- **C++**: Keras model inference in C++ code implemented for CLAS12 de-noiser.

- **Multi-Threading**: Multi-threading implemented to process data files (using std::thread)

![Diagram of multi-threading process]

![Graph showing event processing rate vs number of threads]
De-Noising Results (data)

RUN GROUP-A Pass2 Validation Cooking
Includes De-Nosing and AI-assisted Tracking

$$ep \rightarrow e' p\pi^- (X)$$

Increase 57%

$$ep \rightarrow e' p\pi^+ (X)$$

Increase 59%

$$ep \rightarrow e' \pi^+ \pi^- (X)$$

Increase 67%
De-Noising

CLAS12 Event Display (Drift Chambers)
**Track Parameter Reconstruction**

- **Charge Track Parameter Inference**
  - Reconstruct momentum and angles of particles based on the cluster positions of the tracks.
  - Particles have distinct trajectories through drift chambers depending on their momentum, polar and azimuthal angle.
  - Design an MLP network and investigate different combinations of activation functions to derive the best network for this problem.

- Missing mass of two particles calculated using particle momenta from Hit-Based Tracking compared to missing mass calculated from AI particle parameter inference.

- Hit Based Tracking works ~250 ms per event
- AI reconstructs particle parameters <0.5 ms per event
Future of CLAS12-AI

### Data Acquisition
- **Level-3 Trigger (AI)**
  - | Threshold | Purity | Efficiency | Accuracy |
  - | 0.0012 | 0.841 | 0.9999 | 0.956 |
  - | 0.03 | 0.930 | 0.9999 | 0.962 |
  - | 0.47 | 0.977 | 0.9999 | 0.983 |

### Track Classification (AI)
- Classifying track candidates from reconstructed clusters in real-time.

### Physics Reconstruction (AI)
- Converting hits to clusters at high efficiency.

### Data De-Noising (AI)
- Removing noise signals from tracking detectors.

### Data Persistence
- Saving experimental data already containing tracks and physics topologies identified by AI.
De-nosing Performance With Central Detector

pass1 = 129894
pass2 v1_22/pass1 = 1.618
pass2 v1_24/pass1 = 1.662
pass2 v1_25/pass1 = 1.866

M (eππ) pass 1 cooking
M (eππ) pass 2 cooking
fit [pass1 cooking]
fit [pass2 cooking (ai/da)]

entries: 358809
mean: 0.9457
rms: 0.2218
entries: 634018
mean: 0.9707
rms: 0.1985
p0: 4125.480 ± 26.3898
p1: -20354.307 ± 56.9940
p2: 33311.141 ± 69.1012
p3: -15821.372 ± 52.7093
amp: 1344.296 ± 16.0894
mean: 0.861 ± 0.0009
sigma: 0.069 ± 0.0010
p0: 13973.534 ± 30.2880
p1: -62936.734 ± 66.9614
p2: 91945.890 ± 89.3921
p3: -40164.995 ± 66.6640
amp: 3821.896 ± 24.5976
mean: 0.890 ± 0.0004
sigma: 0.064 ± 0.0005

162841 / 55398 = 2.939