ML Data Compression and Noise Filtering for Real-Time Computing

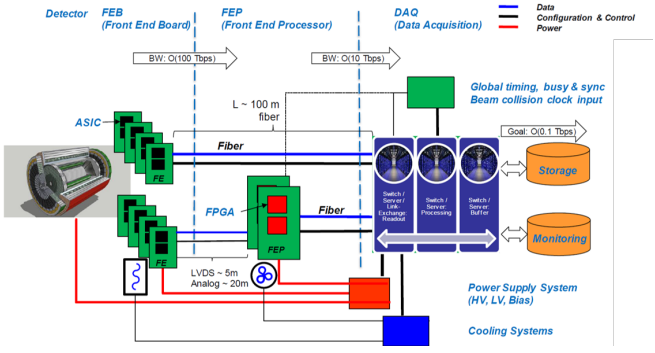
Speaker: Yi Huang[⋆]
Collaborators: Yihui Ren[⋆], Jin Huang[†]

Brookhaven National Laboratory
[⋆]Computational Science Initiative and [†]Physics Department

April 29, 2021

Introduction

Major challenges of Electron-Ion Collision streaming data acquisition

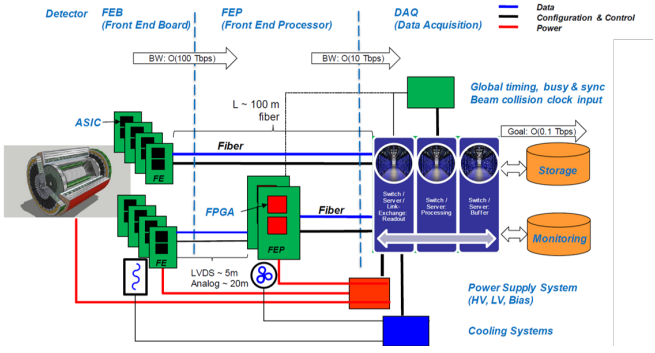


EIC CDR Fig. 8.27: Diagram of the detector readout and DAQ system

- Experiment data may be noisy
- Experiment data can be too large and expensive to fit in persistent storage limit

Introduction

Major challenges of Electron-Ion Collision streaming data acquisition

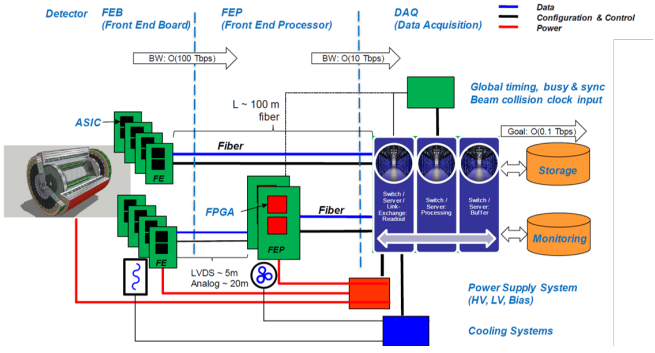


EIC CDR Fig. 8.27: Diagram of the detector readout and DAQ system

- Experiment data may be noisy
- Experiment data can be too large and expensive to fit in persistent storage limit

Introduction

Major challenges of Electron-Ion Collision streaming data acquisition

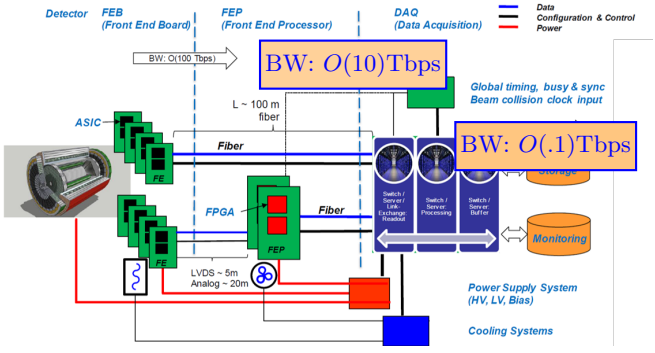


EIC CDR Fig. 8.27: Diagram of the detector readout and DAQ system

- Experiment data may be noisy
- Experiment data can be too large and expensive to fit in persistent storage limit

Introduction

Major challenges of Electron-Ion Collision streaming data acquisition

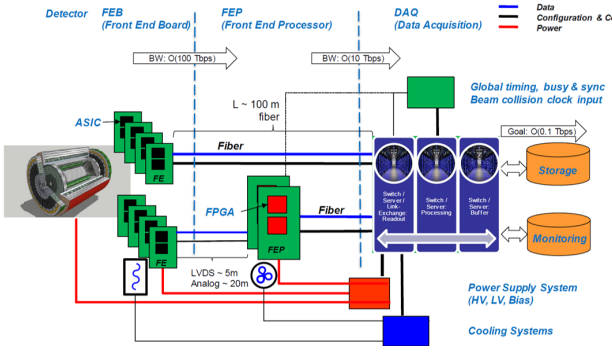


EIC CDR Fig. 8.27: Diagram of the detector readout and DAQ system

- Experiment data may be noisy
- Experiment data can be too large and expensive to fit in persistent storage limit

Introduction

Major challenges of Electron-Ion Collision streaming data acquisition



EIC CDR Fig. 8.27: Diagram of the detector readout and DAQ system

- Experiment data may be noisy
- Experiment data can be too large and expensive to fit in persistent storage limit

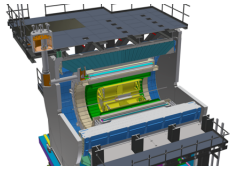
Goal

Using machine learning for **data compression** and **noise filtering**.

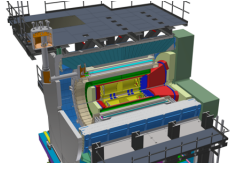
Introduction

Time projection chamber (TPC) data

- Time projection chamber is a popular choice of main tracking detector for both RHIC and EIC experiments.
- **Compression**: TPC data dominates the data volume
- **Noise filtering**: TPC data may contain a high amount of noise ($> 50\%$) from the experiment background
- **High throughput** to match TPC data taking



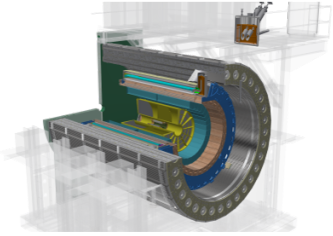
sPHENIX @ RHIC, 2023-2025
`https://indico.mit.edu/event/1/contributions/73/`



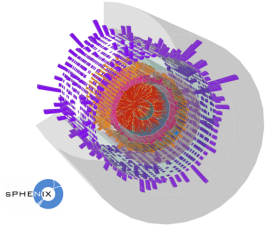
One of the EIC detector concepts, $\sim$2030
`https://indico.mit.edu/event/1/contributions/75/`

Introduction

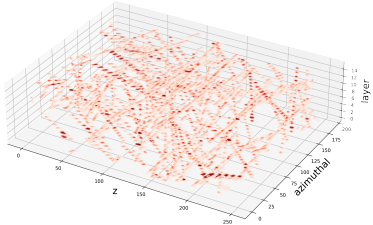
TPC data in this study



Detector model



Detector simulation



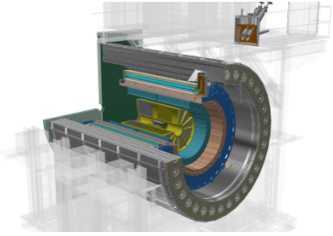
An example of TPC data frame

Preparing for the toughest

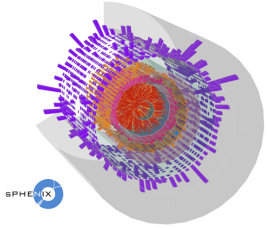
In this study, we use the 10% central Au + Au collision with 170kHz pile up, which is busiest event in sPHENIX.

Introduction

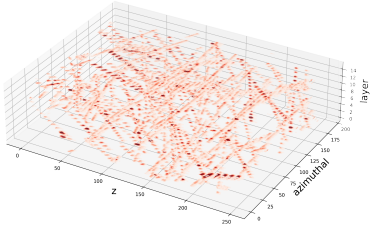
TPC data in this study



Detector model



Detector simulation

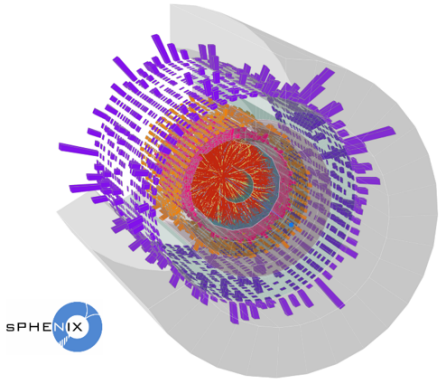


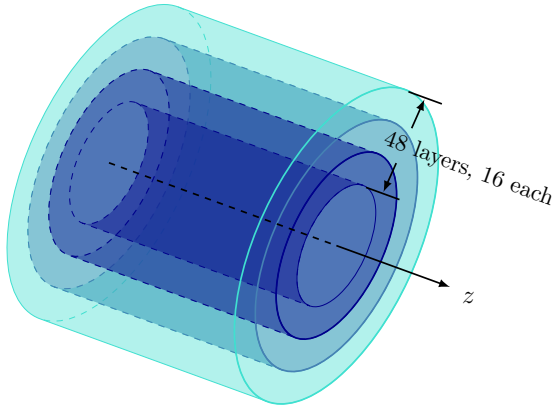
An example of TPC data frame

Preparing for the toughest

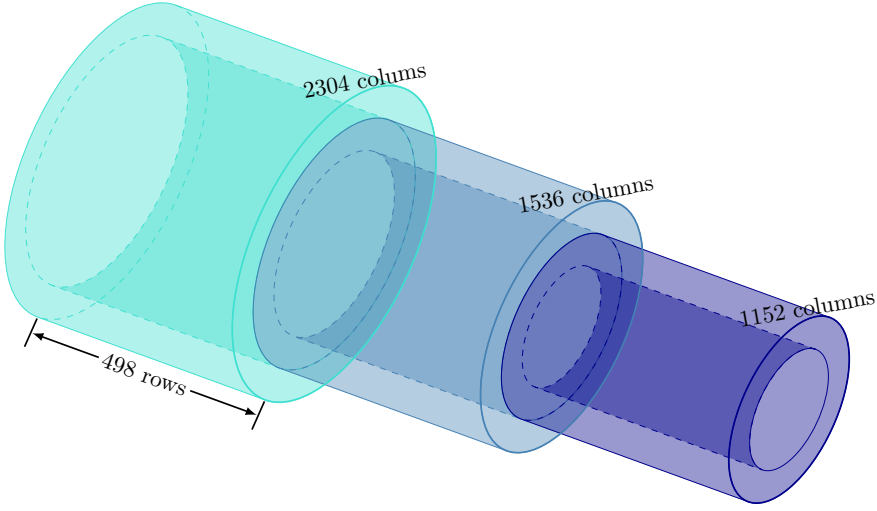
In this study, we use the 10% central Au + Au collision with 170kHz pile up, which is busiest event in sPHENIX.

Time projection chamber zoom-in





Time projection chamber zoom-in



The Amount of Data Generated by TPC

- **Data format**: 10-bit integer (ADC value range $\in [0, 1023]$)
- **Number of voxels**: (azimuth $\times$ z $\times$ layer)
  - Outer layer group: $2304 \times 498 \times 16 \approx 18$M;
  - Middle layer group: $1536 \times 498 \times 16 \approx 12$M;
  - Inner layer group: $1152 \times 498 \times 16 \approx 9$M
- **Digitization frequency**: 20MHz;
  Frame Frequency: 80KHz

Uncompressed data rate: $\sim$30 Tera bits per second

Average compressed data rate via SAMPA ASIC: $\sim$ 2Tbps

[Thursday morning talk by Takao Sakaguchi]

The Amount of Data Generated by TPC

- **Data format**: 10-bit integer (ADC value range $\in [0, 1023]$)
- **Number of voxels**: (azimuth $\times$ z $\times$ layer)
  - Outer layer group: $2304 \times 498 \times 16 \approx 18$M;
  - Middle layer group: $1536 \times 498 \times 16 \approx 12$M;
  - Inner layer group: $1152 \times 498 \times 16 \approx 9$M
- **Digitization frequency**: 20MHz;
  Frame Frequency: 80KHz

Uncompressed data rate: $\sim$30 Tera bits per second

Average compressed data rate via SAMPA ASIC: $\sim$ 2Tbps

[Thursday morning talk by Takao Sakaguchi]

The Amount of Data Generated by TPC

- **Data format**: 10-bit integer (ADC value range $\in [0, 1023]$)
- **Number of voxels**: (azimuth $\times$ z $\times$ layer)
  - Outer layer group: $2304 \times 498 \times 16 \approx 18$M;
  - Middle layer group: $1536 \times 498 \times 16 \approx 12$M;
  - Inner layer group: $1152 \times 498 \times 16 \approx 9$M
- **Digitization frequency**: 20MHz;
  Frame Frequency: 80KHz

Uncompressed data rate: $\sim$30 Tera bits per second

Average compressed data rate via SAMPA ASIC: $\sim$ 2Tbps

[Thursday morning talk by Takao Sakaguchi]

Lossy Compression Algorithms

There are many existing compression algorithms designed for simulation-heavy scientific data represented by dense matrices of high-precision floating-point values.

- SZ: Error-bounded lossy compressor for HPC data
  https://github.com/szcompressor/SZ
- ZFP: Compressor for integer and floating-point data stored in multidimensional arrays
  https://github.com/LLNL/zfp
- MGARD: MultiGrid adaptive reduction of data
  https://github.com/CODARcode/MGARD

Problems with existing compressors

Hand-crafted and manually-tuned to suit data, missing learnable noise filtering.

Lossy Compression Algorithms

There are many existing compression algorithms designed for simulation-heavy scientific data represented by dense matrices of high-precision floating-point values.

- SZ: Error-bounded lossy compressor for HPC data
  https://github.com/szcompressor/SZ
- ZFP: Compressor for integer and floating-point data stored in multidimensional arrays
  https://github.com/LLNL/zfp
- MGARD: MultiGrid adaptive reduction of data
  https://github.com/CODARcode/MGARD

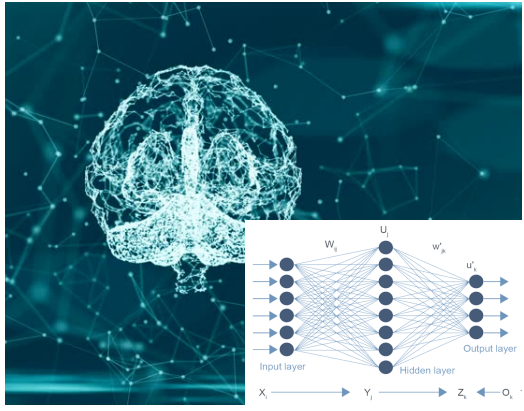
Problems with existing compressors

Hand-crafted and manually-tuned to suit data, missing learnable noise filtering.

Convolutional Neural Encoder

What is that and why we think it should work

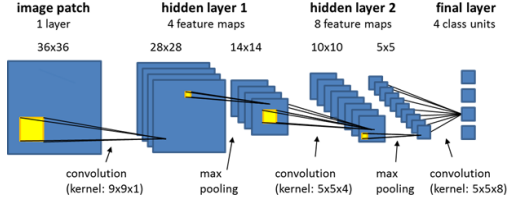
- Artificial neural network (ANN)
  (an ANN helps machine learn a function just as a nervous system does for a living organism)
- Convolutional neural network
  (an ANN architecture that can handle high volume image data)
- Auto encoder
  (an ANN encoder learns its own encoding rule with the help from a ANN decoder)



Convolutional Neural Encoder

What is that and why we think it should work

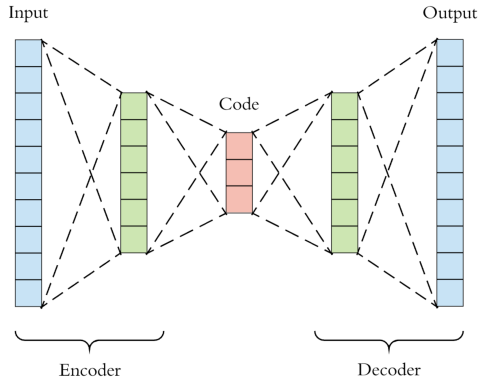
- Artificial neural network (ANN)
  (an ANN helps machine learn a function just as a nervous system does for a living organism)
- Convolutional neural network
  (an ANN architecture that can handle high volume image data)
- Auto encoder
  (an ANN encoder learns its own encoding rule with the help from a ANN decoder)



Convolutional Neural Encoder

What is that and why we think it should work

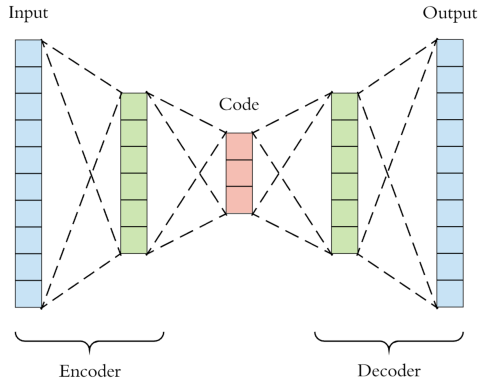
- Artificial neural network (ANN)
  (an ANN helps machine learn a function just as a nervous system does for a living organism)
- Convolutional neural network
  (an ANN architecture that can handle high volume image data)
- Auto encoder
  (an ANN encoder learns its own encoding rule with the help from a ANN decoder)



Convolutional Neural Encoder

What is that and why we think it should work

- Artificial neural network (ANN)
  (an ANN helps machine learn a function just as a nervous system does for a living organism)
- Convolutional neural network
  (an ANN architecture that can handle high volume image data)
- Auto encoder
  (an ANN encoder learns its own encoding rule with the help from a ANN decoder)

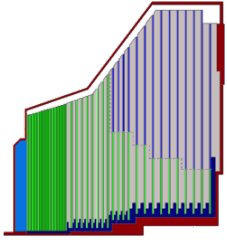


Desirable properties of a neural encoder

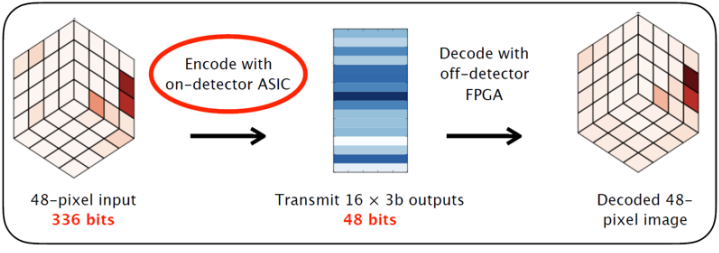
Data-driven coding rule to optimize domain specific tasks, such as noise filtering.

Example of on-going auto-encoder study in modern data acquisition

Auto-encode evaluated for on-detector data compression for CMS HGC
[Reference to talk: https://indico.fnal.gov/event/46746/contributions/210450/]



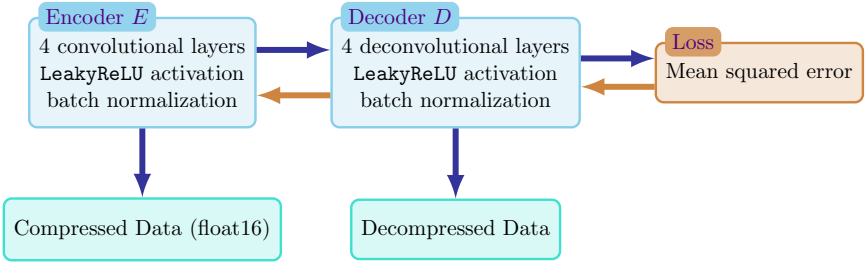
Compact Muon Solenoid
High-Granularity Calorimeter



Proposed data flow with auto-encoder on
application-specific integrated circuit

Convolutional Neural Encoder

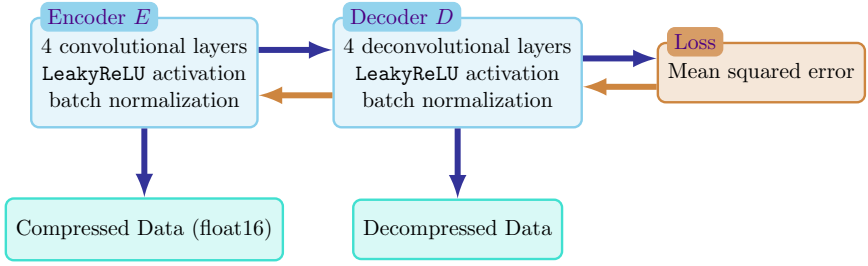
A basic idea



- The encoder network E compresses the data;
- The decoder network D decompresses the compressed data;
- The encoder and decoder are trained in an end-to-end fashion.
- Suitable for training with real data.

Convolutional Neural Encoder

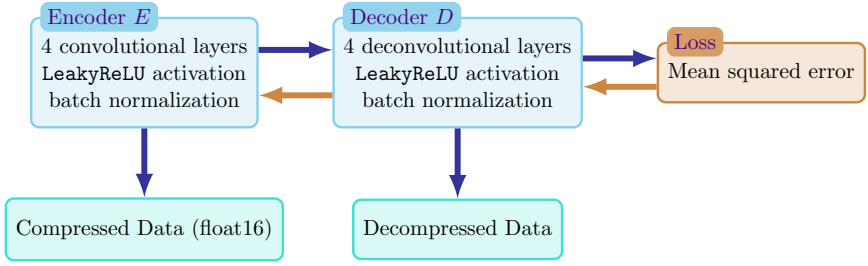
A basic idea



- The encoder network E compresses the data;
- The decoder network D decompresses the compressed data;
- The encoder and decoder are trained in an end-to-end fashion.
- Suitable for training with real data.

Convolutional Neural Encoder

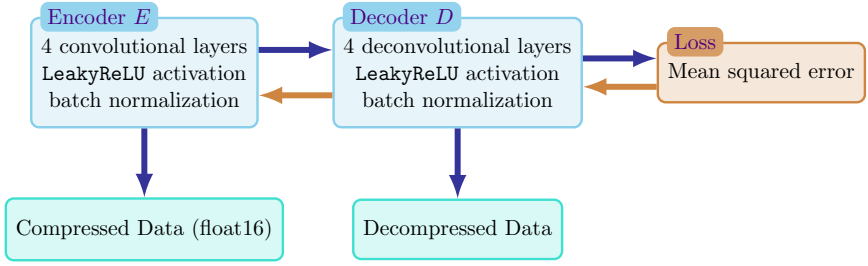
A basic idea



- The encoder network E compresses the data;
- The decoder network D decompresses the compressed data;
- The encoder and decoder are trained in an end-to-end fashion.
- Suitable for training with real data.

Convolutional Neural Encoder

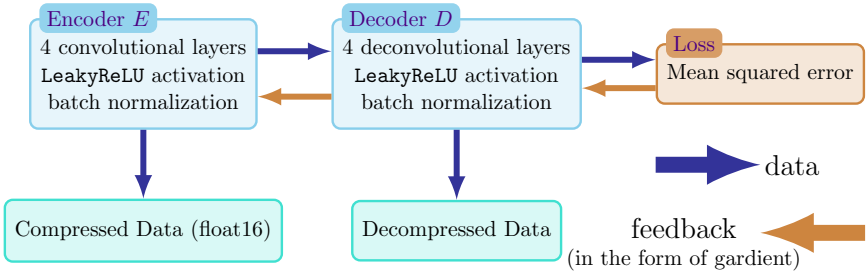
A basic idea



- The encoder network E compresses the data;
- The decoder network D decompresses the compressed data;
- The encoder and decoder are trained in an end-to-end fashion.
- Suitable for training with real data.

Convolutional Neural Encoder

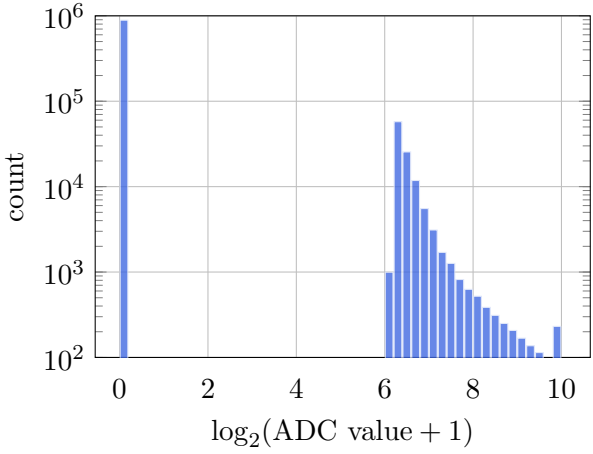
A basic idea



- The encoder network E compresses the data;
- The decoder network D decompresses the compressed data;
- The encoder and decoder are trained in an end-to-end fashion.
- Suitable for training with real data.

Convolutional Neural Encoder

Problem with the basic idea



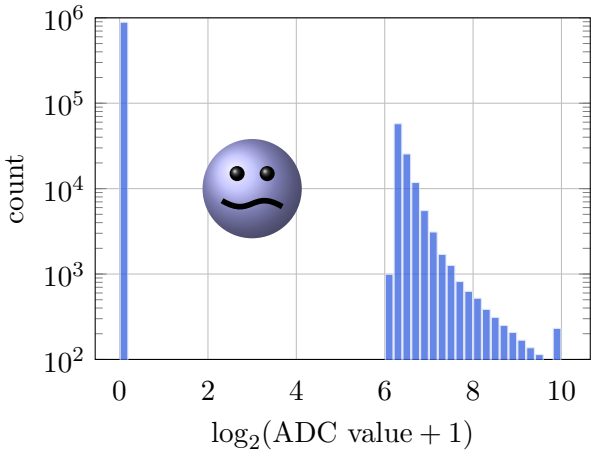
The distribution:

- is bi-modal
- is unbalanced
- is skewed (having a sharp edge at 6)
- has a long and slender tail

Not the most friendly to NN

Convolutional Neural Encoder

Problem with the basic idea



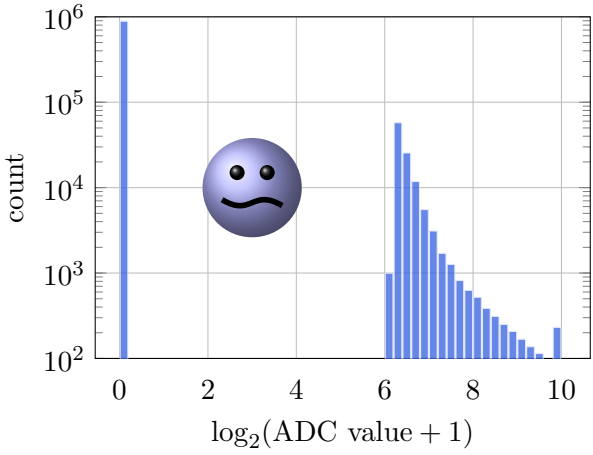
The distribution:

- is bi-modal
- is unbalanced
- is skewed (having a sharp edge at 6)
- has a long and slender tail

Not the most friendly to NN

Convolutional Neural Encoder

Problem with the basic idea



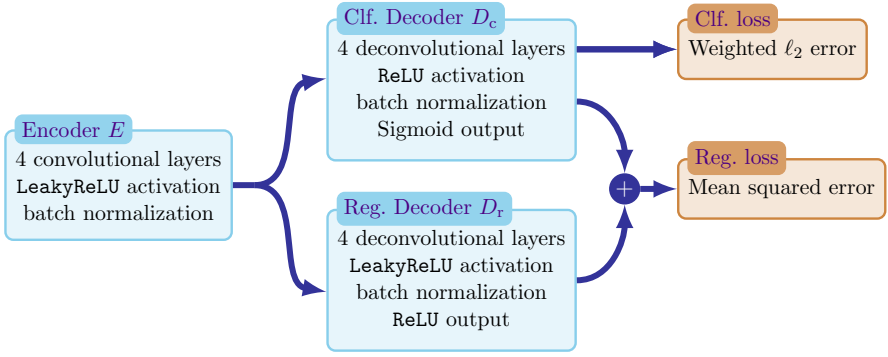
The distribution:

- is bi-modal
- is unbalanced
- is skewed (having a sharp edge at 6)
- has a long and slender tail

Not the most friendly to NN

Convolutional Neural Encoder

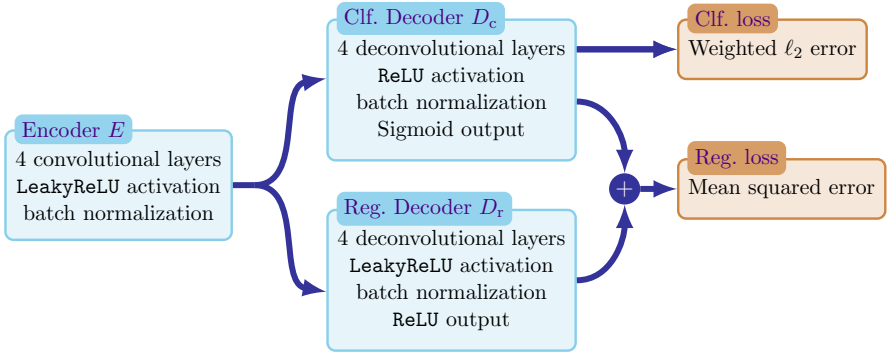
A better solution: double decoders



- Classification decoder D_c learns to recognize truth signal
- Regression decoder D_r learns to approximate the value of truth signal
- Decompressed data = regression masked by classification $\Rightarrow$ **Noise Filtering**

Convolutional Neural Encoder

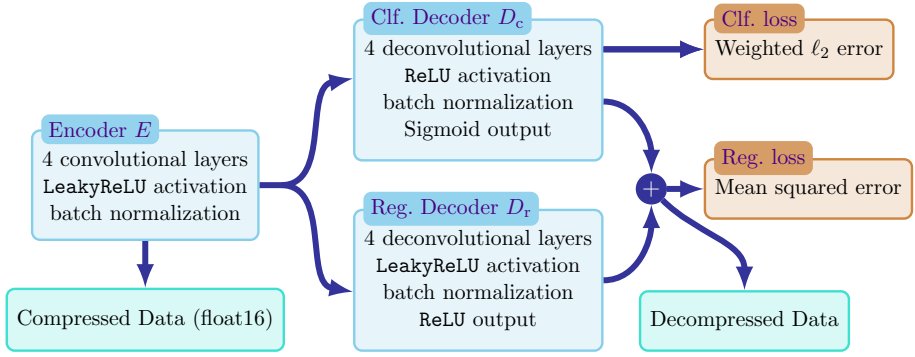
A better solution: double decoders



- Classification decoder D_c learns to recognize truth signal
- Regression decoder D_r learns to approximate the value of truth signal
- Decompressed data = regression masked by classification ⇒ **Noise Filtering**

Convolutional Neural Encoder

A better solution: double decoders

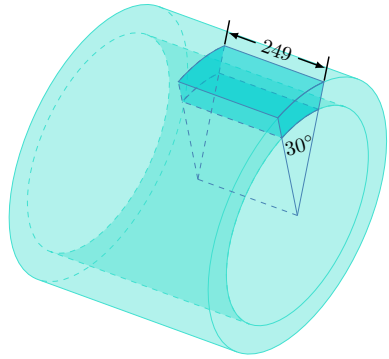


- Classification decoder D_c learns to recognize truth signal
- Regression decoder D_r learns to approximate the value of truth signal
- Decompressed data = regression masked by classification $\Rightarrow$ **Noise Filtering**

Convolutional Neural Encoder

Input

- a 30° degree sector along the azimuth direction (192 columns for the outer layer group)
- a half the z-direction (249 rows)
- one layer group (16 layers)



Convolutional Neural Encoder

Results I: Compression ratio and mean squared error

- Compression ratio is $1:27$
  (1 : 3 for ASIC for this busiest event)
- Mean squared error $\approx 1600\text{ADU}^2$

Convolutional Neural Encoder

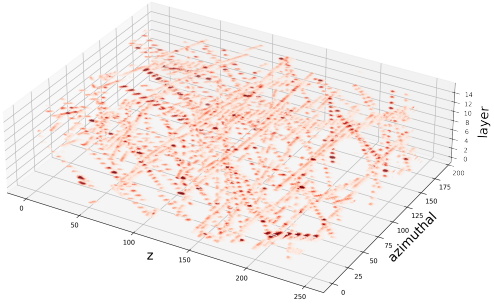
Results I: Compression ratio and mean squared error

- Compression ratio is $1:27$
  (1 : 3 for ASIC for this busiest event)
- Mean squared error $\approx 1600\text{ADU}^2$
  MSE is still quite large. We need to do more study on how to adjust the network to handle data with a sharp zero suppression cut-off $\Rightarrow$ expect improved MSE.

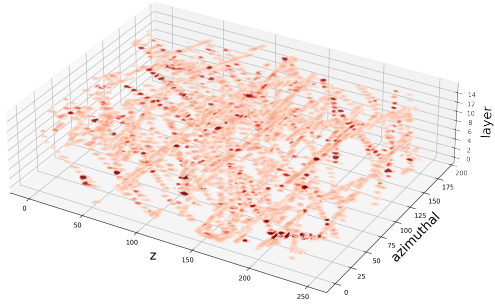
Convolutional Neural Encoder

Results II: 3d original v.s. decompressed

Original



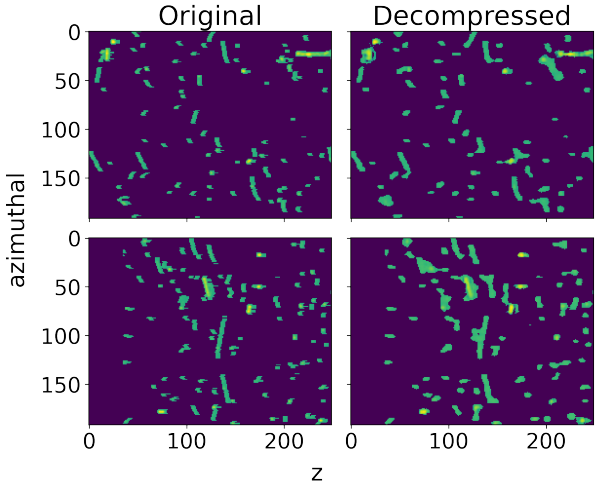
Decompressed



Global feature is well reproduced. Local variations are still to be quantified in downstream analysis.

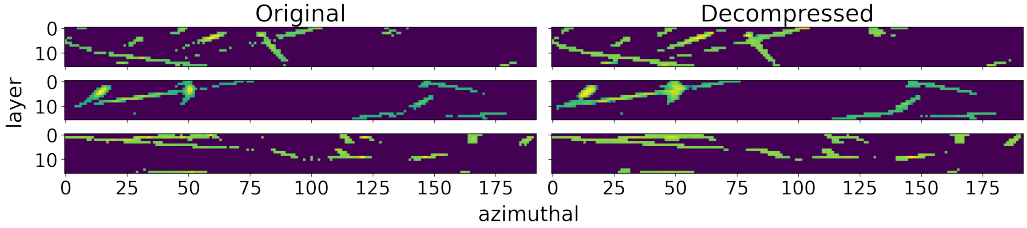
Convolutional Neural Encoder

Results III: 2d sections original v.s. decompressed

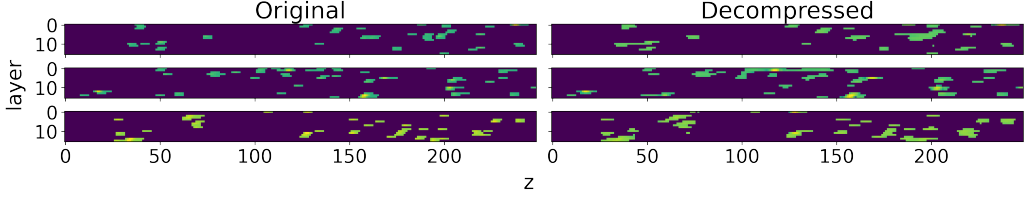


Convolutional Neural Encoder

Results III: 2d sections original v.s. decompressed, cont.



Prominent patterns in azimuth-layer slice are corresponding to curved low momentum track.



Summary and Future Direction

- Testing auto-encoder-based compression and noise filtering network on highest occupancy TPC data.
- Reach $1:27$ compression ratio while preserve the general features.
- Future directions:
  - Optimizing the depth of the network
  - Optimizing the shape of the CNN kernels
  - Integrating simulation ground truth into the training to improve noise rejection.
  - Working well for downstream applications (for example: clustering and tracking efficiency and position resolution)
  - Data acquisition hardware integration

Summary and Future Direction

- Testing auto-encoder-based compression and noise filtering network on highest occupancy TPC data.
- Reach $1:27$ compression ratio while preserve the general features.
- Future directions:
  - Optimizing the depth of the network
  - Optimizing the shape of the CNN kernels
  - Integrating simulation ground truth into the training to improve noise rejection.
  - Working well for downstream applications (for example: clustering and tracking efficiency and position resolution)
  - Data acquisition hardware integration