Real-time data compression with Bicephalous Convolutional Auto-Encoder

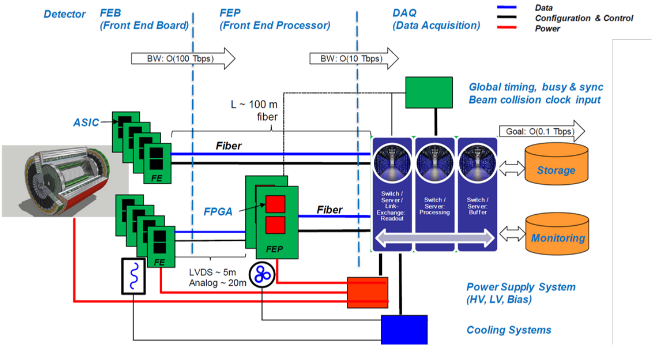
Speaker: Yi Huang[⋆]
Collaborators: Yihui Ren[⋆], Jin Huang[†]

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

Sept. 9, 2021

Introduction

Major challenges of Electron-Ion Collision streaming data acquisition

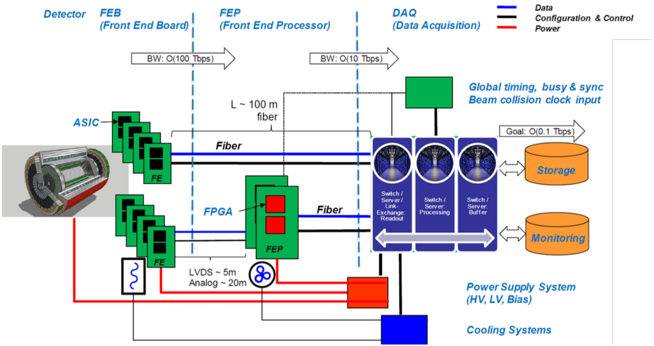


EIC CDR Fig. 8.27: Diagram of the detector readout and DAQ system [Ref. "EIC readout overview" by Fernando Barbosa]

- EIC signal data rate is low and we aim to stream readout all variety of collision signal
- Experiment data may be noisy and filled with background hits
- Experiment data can be too large and expensive to fit in persistent storage limit

Introduction

Major challenges of Electron-Ion Collision streaming data acquisition

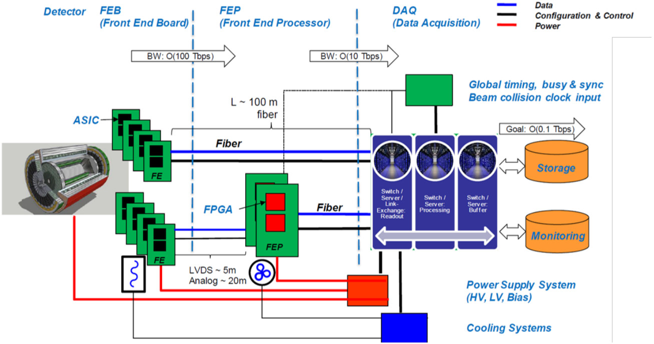


EIC CDR Fig. 8.27: Diagram of the detector readout and DAQ system [Ref. “EIC readout overview” by Fernando Barbosa]

- EIC signal data rate is low and we aim to stream readout all variety of collision signal
- Experiment data may be noisy and filled with background hits
- Experiment data can be too large and expensive to fit in persistent storage limit

Introduction

Major challenges of Electron-Ion Collision streaming data acquisition

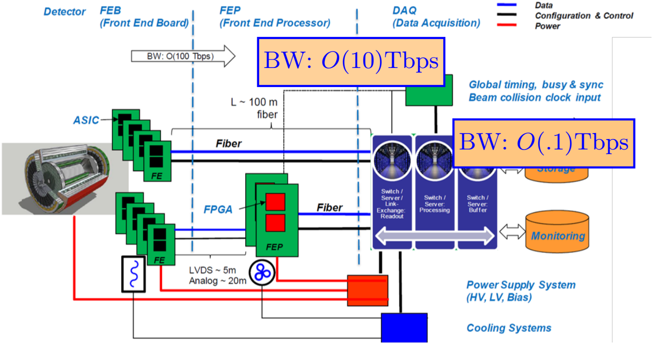


EIC CDR Fig. 8.27: Diagram of the detector readout and DAQ system [Ref. "EIC readout overview" by Fernando Barbosa]

- EIC signal data rate is low and we aim to stream readout all variety of collision signal
- Experiment data may be noisy and filled with background hits
- Experiment data can be too large and expensive to fit in persistent storage limit

Introduction

Major challenges of Electron-Ion Collision streaming data acquisition

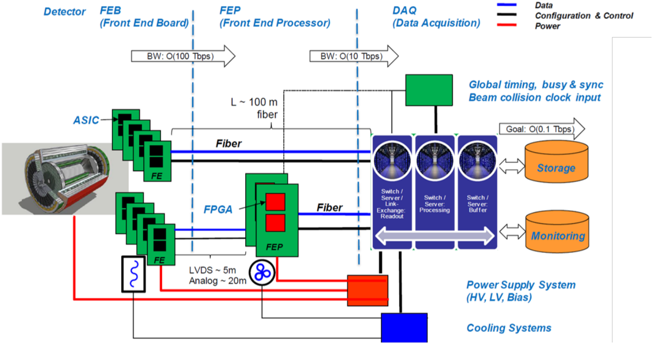


EIC CDR Fig. 8.27: Diagram of the detector readout and DAQ system [Ref. "EIC readout overview" by Fernando Barbosa]

- EIC signal data rate is low and we aim to stream readout all variety of collision signal
- Experiment data may be noisy and filled with background hits
- 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 [Ref. "EIC readout overview" by Fernando Barbosa]

- EIC signal data rate is low and we aim to stream readout all variety of collision signal
- Experiment data may be noisy and filled with background hits
- Experiment data can be too large and expensive to fit in persistent storage limit

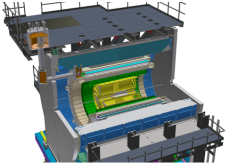
Goal

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

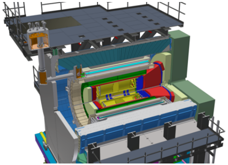
Introduction

Time projection chamber (TPC) data

- Time projection chamber is a popular choice of main tracking detector for both RHIC and EIC experiments
  - Using the sPHENIX TPC data model for this study: high data rate and well modeled device
  - Algorithm would be applicable for EIC tracker and calorimeter too
- **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



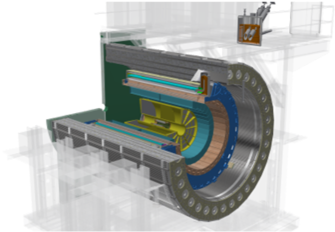
sPHENIX @ RHIC, 2023-2025
sPHENIX Technical Design Report



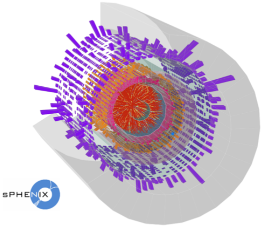
One of the EIC detector concepts, ∼2030
arXiv:1402.1209

Introduction

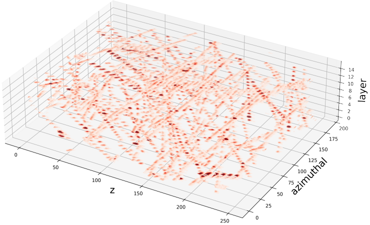
TPC data in this study



Detector model



Detector simulation



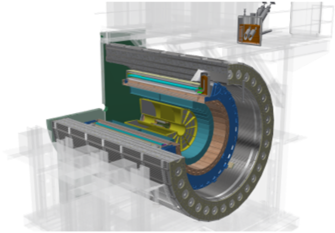
An example of
TPC data frame

Preparing for the toughest

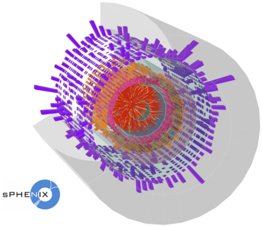
In this study, we use the 10% central Au + Au collision with 170kHz pile up, which is busiest event in sPHENIX.

Introduction

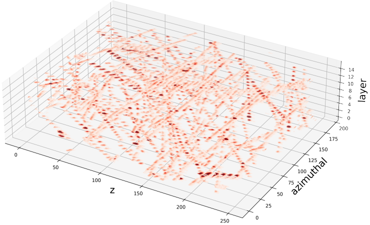
TPC data in this study



Detector model



Detector simulation

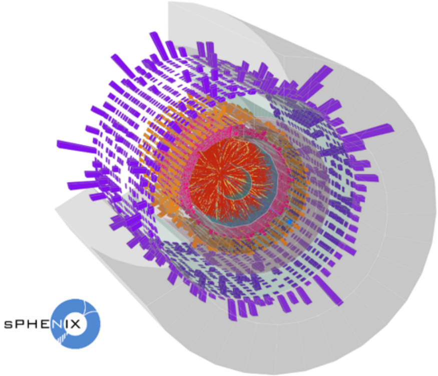


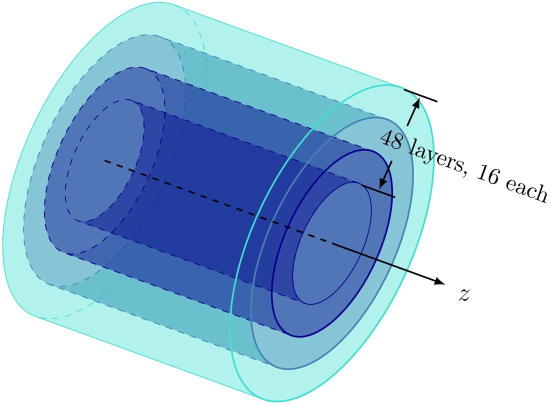
An example of
TPC data frame

Preparing for the toughest

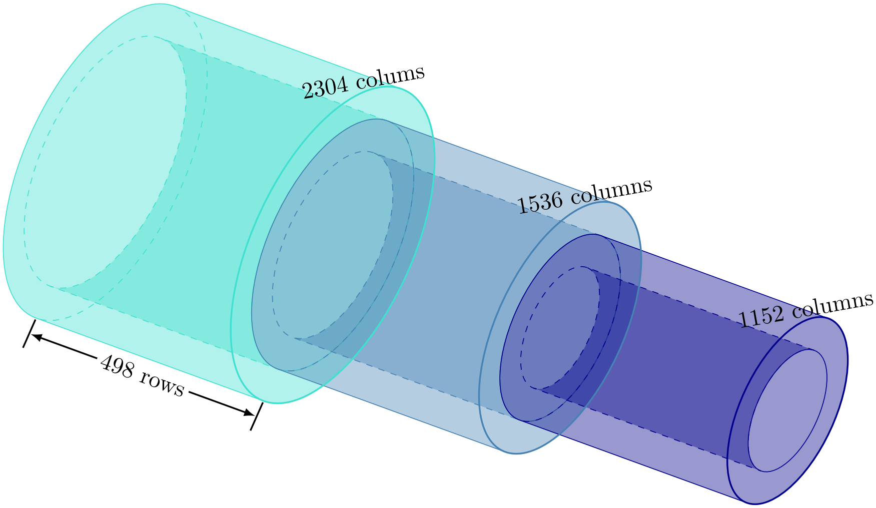
In this study, we use the 10% central Au + Au collision with 170kHz pile up, which is busiest event in sPHENIX.

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: $2304 \times 498 \times 16 \approx 18M$;
  - Middle layer group: $1536 \times 498 \times 16 \approx 12M$;
  - Inner layer group: $1152 \times 498 \times 16 \approx 9M$
- **Digitization frequency**: 20MHz;
  Frame Frequency: 80KHz

Uncompressed data rate: ~30 Tera bits per second

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

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

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

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

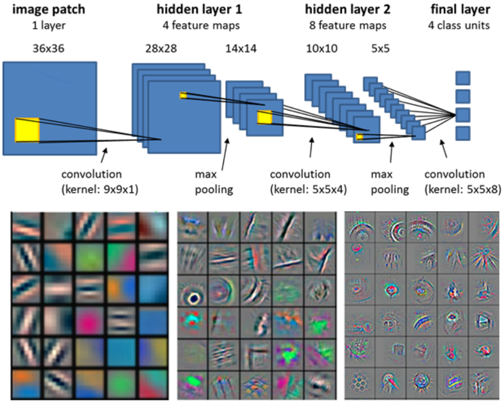
Problems with existing compressors

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

Convolutional Neural Encoder

Why we think it should work

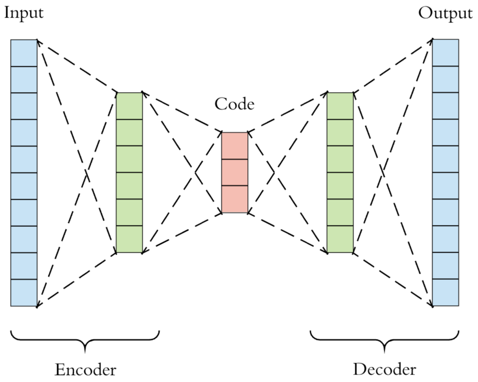
- Convolutional neural network
  (an NN architecture specialized in processing high volume image data)
- Auto encoder
  (an NN encoder learns its own encoding rule with the help from an NN decoder)



Convolutional Neural Encoder

Why we think it should work

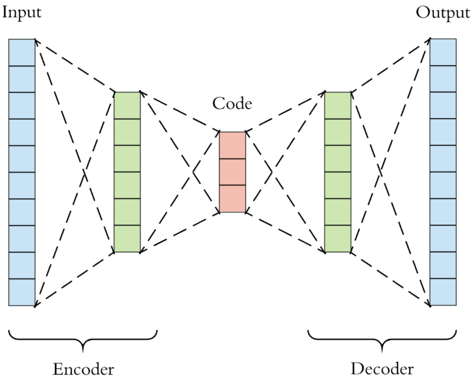
- Convolutional neural network
  (an NN architecture specialized in processing high volume image data)
- Auto encoder
  (an NN encoder learns its own encoding rule with the help from an NN decoder)



Convolutional Neural Encoder

Why we think it should work

- Convolutional neural network
  (an NN architecture specialized in processing high volume image data)
- Auto encoder
  (an NN encoder learns its own encoding rule with the help from an NN decoder)

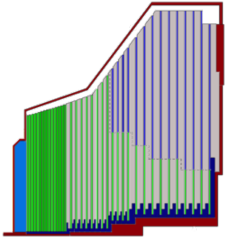


Desirable properties of a neural encoder

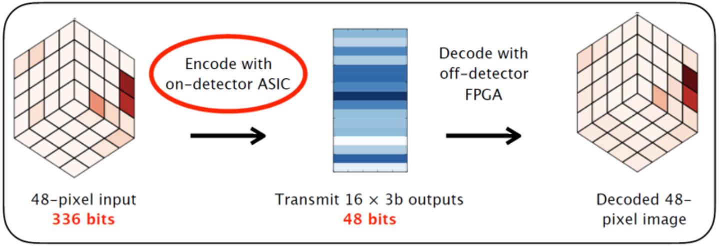
Data-driven coding rule to optimize domain specific tasks, such as noise filtering.

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/]



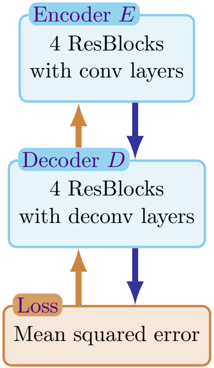
Compact Muon Solenoid High-Granularity Calorimeter

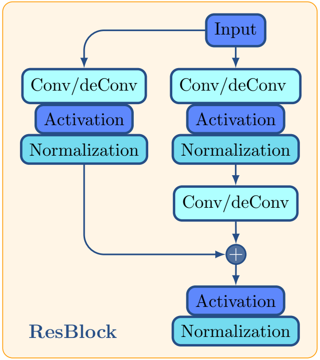


Proposed data flow with auto-encoder on application-specific integrated circuit

Convolutional Neural Encoder

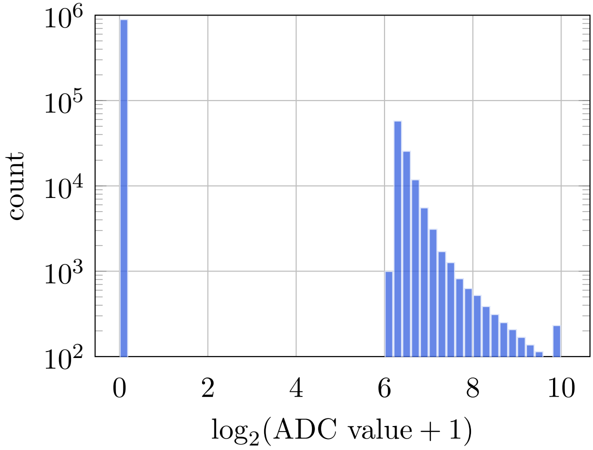
A basic idea





Convolutional Neural Encoder

Problem with the basic idea

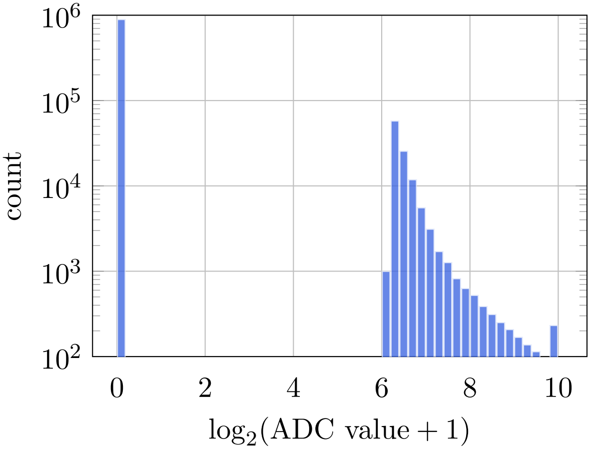


The distribution:

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

Convolutional Neural Encoder

Problem with the basic idea

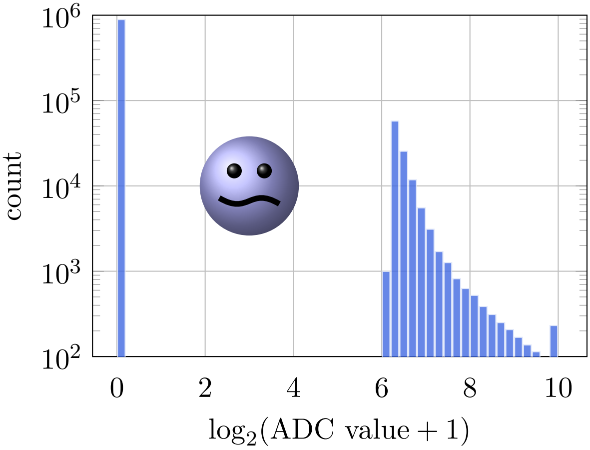


The distribution:

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

Convolutional Neural Encoder

Problem with the basic idea

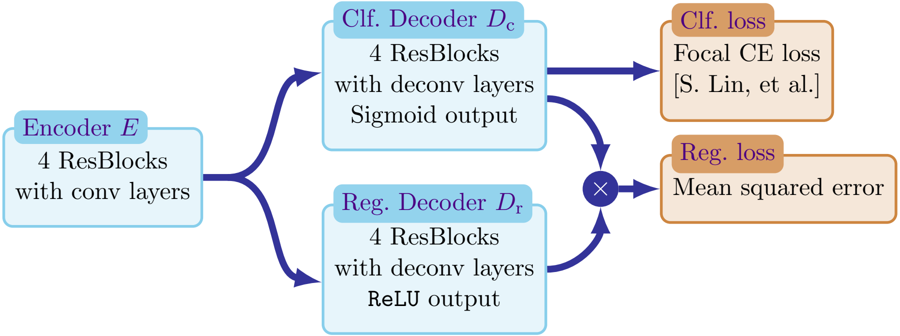


The distribution:

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

Bicephalous Convolutional Neural Encoder

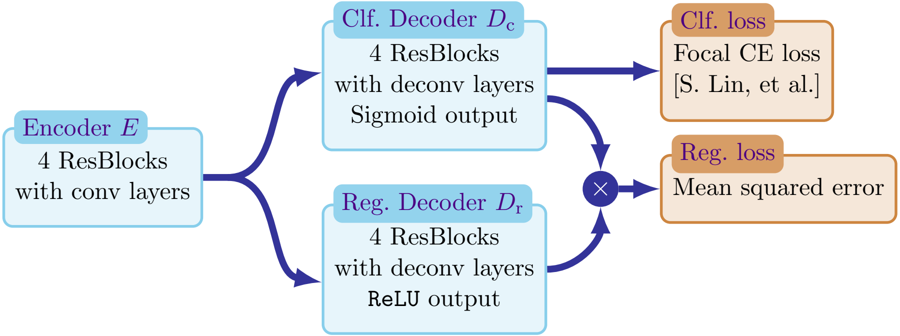
Two heads is better than one



- 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

Bicephalous Convolutional Neural Encoder

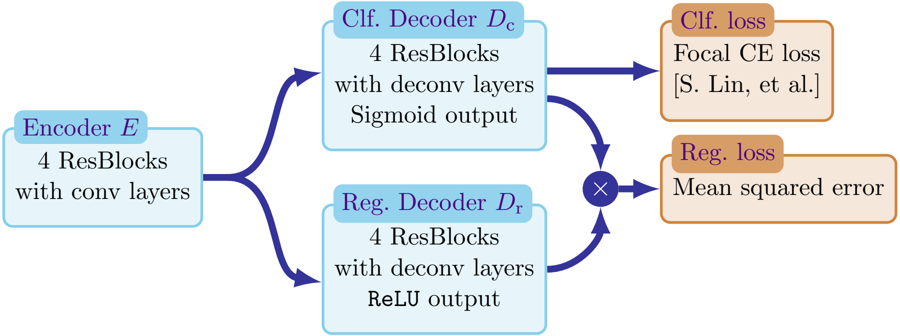
Two heads is better than one



- 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

Bicephalous Convolutional Neural Encoder

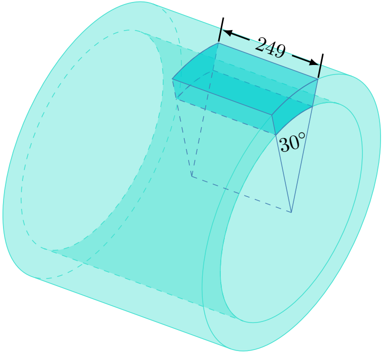
Two heads is better than one



- 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

Input

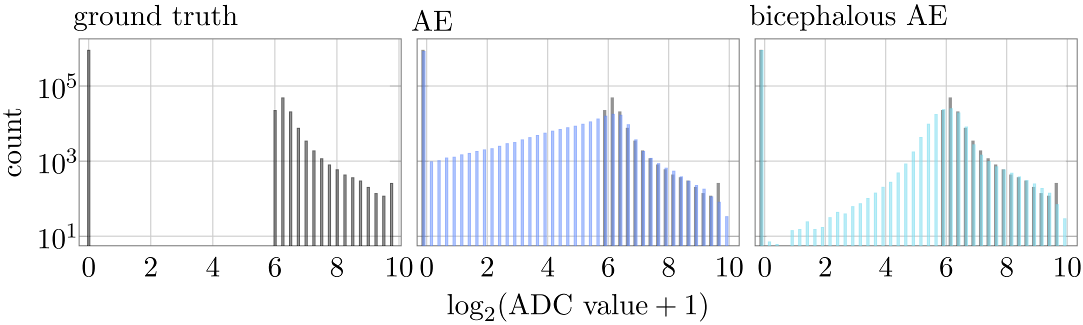
- a 30° degree sector along the azimuth direction (192 columns for the outer layer group)
- a half the z-direction (249 rows)
- one layer group (16 layers)



Results I: AE v.s. Bicephalous AE

Compression ratio is 1 : 27

(1 : 3 for SAMPA ASIC for this busiest event)

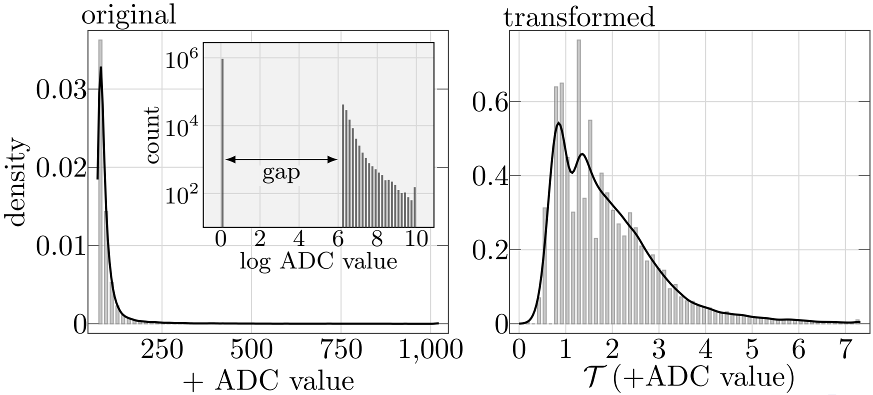


A Missing Ingredient – Input Transform

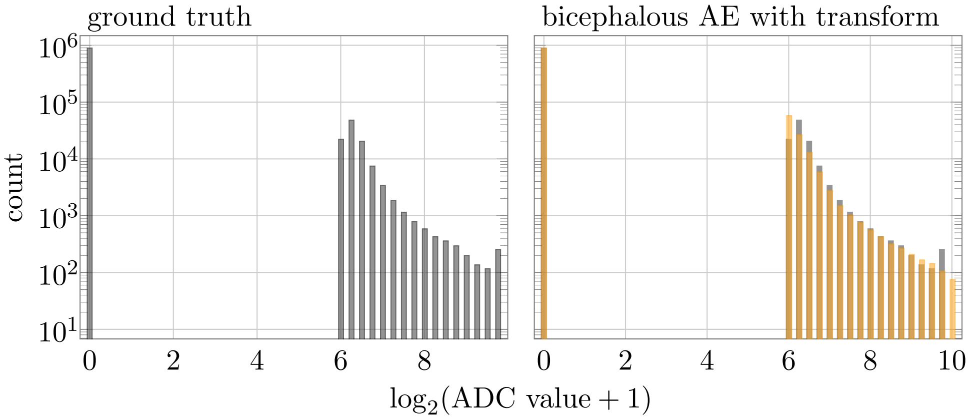
borrowed an idea from [Y. Alanazi, et al.]

$$\text{Input transform: } \mathcal{T}(x) = \log(x-64)/6, \qquad x > 64$$

$$\text{Inverse transform: } \mathcal{T}^{-1}(y) = 64 + \exp(6y), \qquad x \in \mathbb{R}$$



Results II. Bicephalous AE with Input Transform

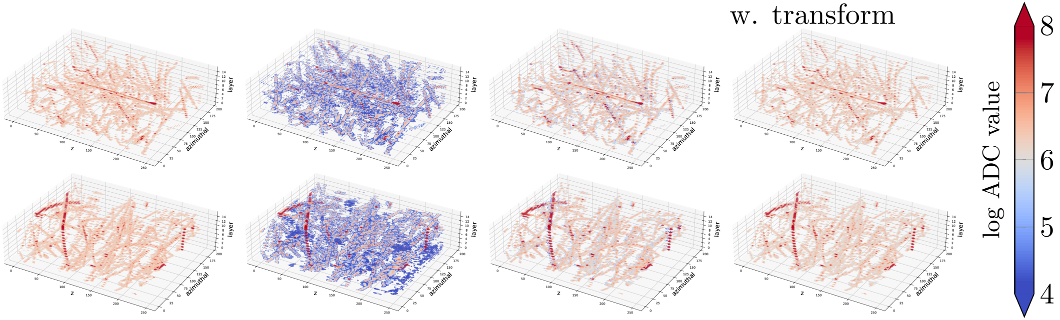


Result III. Ablation Study

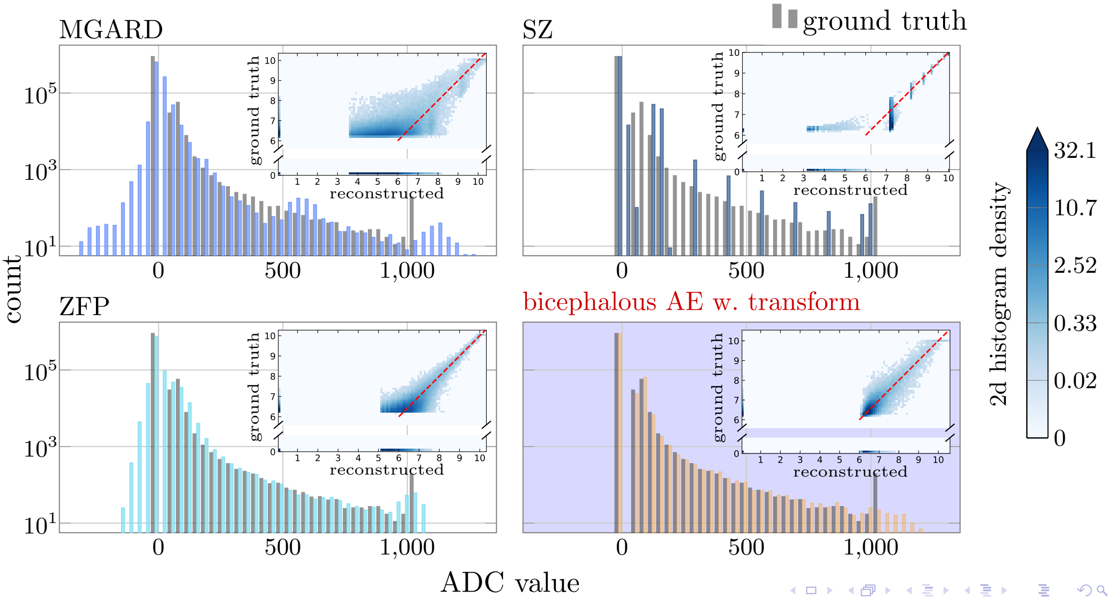
ground truth | AE | bicephalous AE | bicephalous AE w. transform

example 1

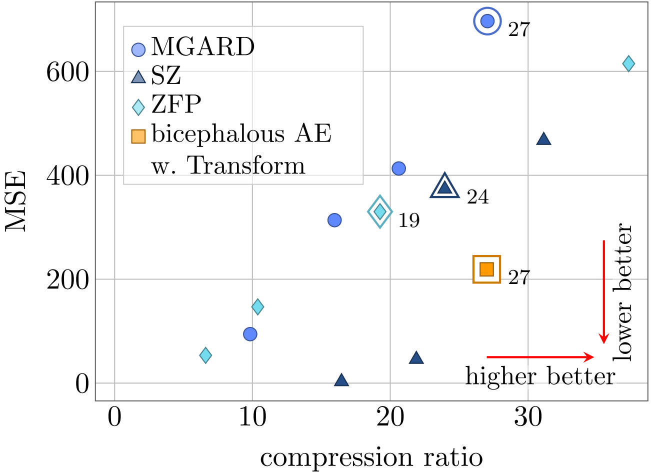
example 2



Result IV-i. Comparing with Existing Compression Algorithms



Result IV-ii. Comparing with Existing Compression Algorithms



Result V. Metrics Summary

Table: Performance comparison

	Compr. ratio↑	MSE↓	log MAE↓	PSNR↑
MGARD	**27**	626.28	1.213	3.223
SZ	24	369.69	0.302	3.452
ZFP	19	219.48	0.267	3.678
AE	**27**	227.61	0.349	3.703
Bicephalous AE	**27**	230.59	0.193	3.706
Bicephalous AE w. Transform	**27**	**218.33**	**0.185**	**3.724**

Summary and Future Direction

- Test auto-encoder-based compression and noise filtering network on highest occupancy TPC data.
- Reach $1:27$ compression ratio.
- Future directions:
  - 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

- Test auto-encoder-based compression and noise filtering network on highest occupancy TPC data.
- Reach $1:27$ compression ratio.
- Future directions:
  - 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