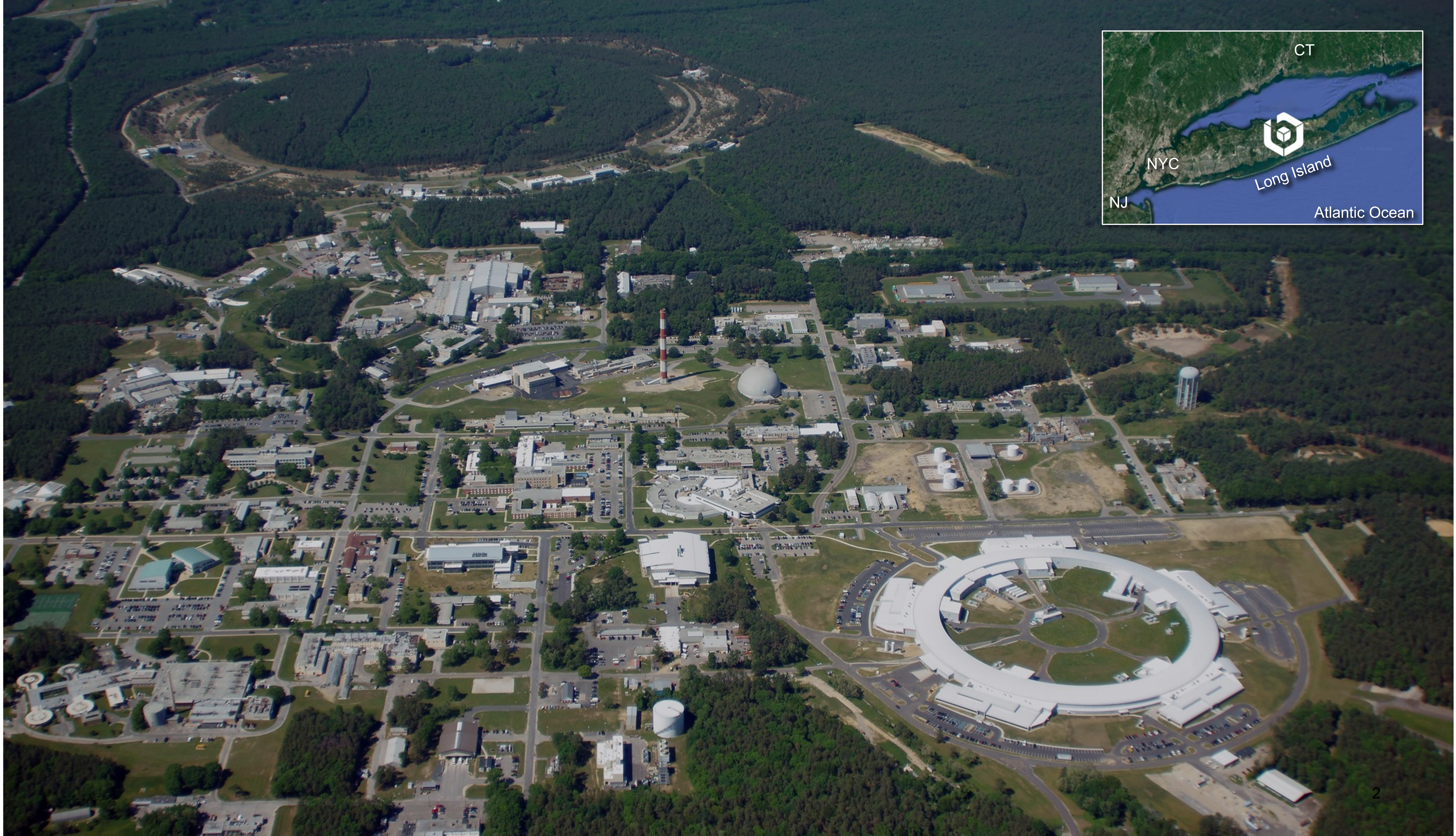
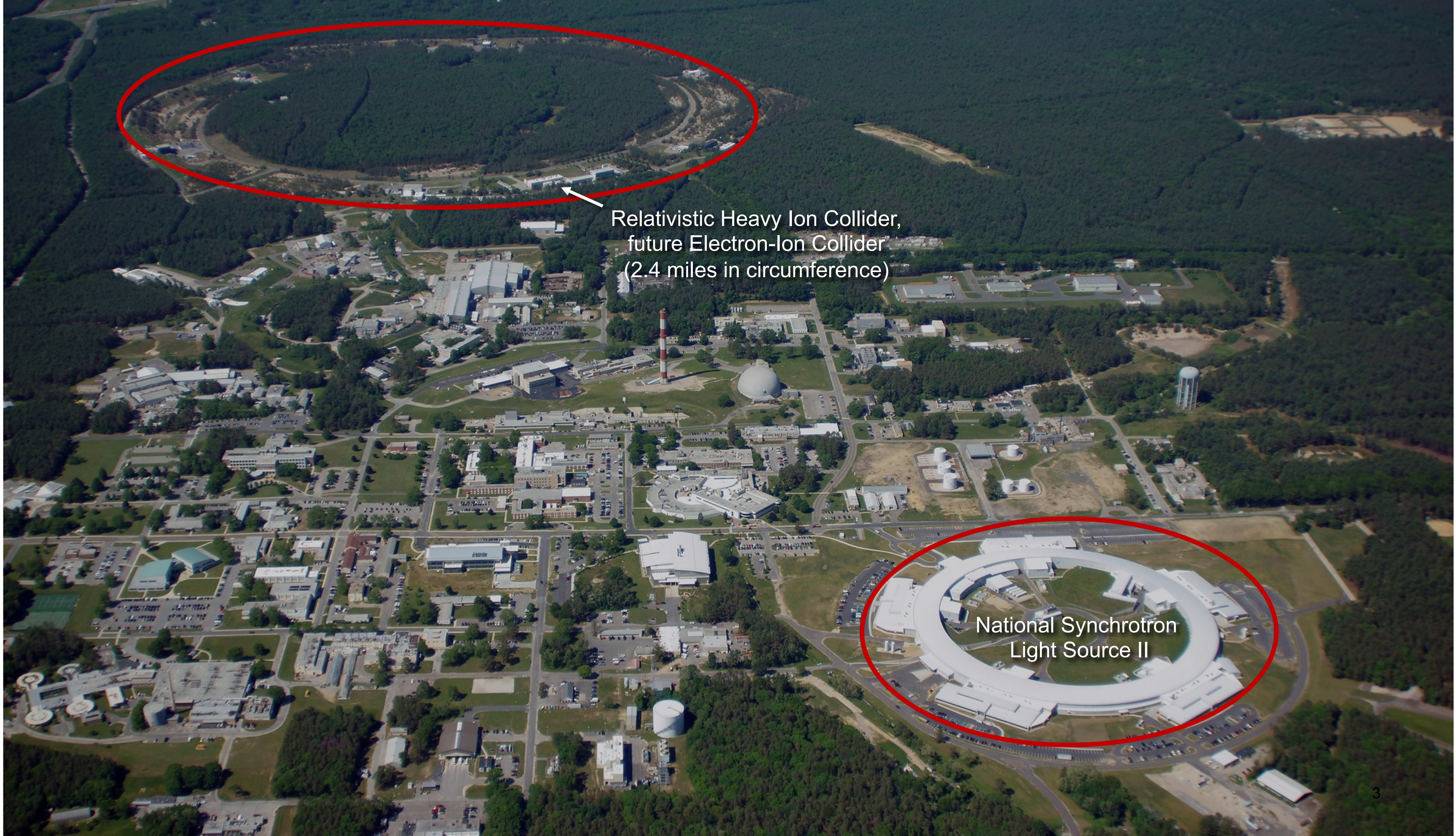
Neural Compression for sPHENIX sparse TPC Data

Yi Huang, Yihui “Ray” Ren, Yeonju Go, Xihaier Luo, Shuhang Li, Thomas Marshall, Joseph D. Osborn, Christopher Pinkenburg, Evgeny Shulga, Shinjae Yoo, Byung-Jun Yoon, Jin Huang (PI)

Presenter: Yihui “Ray” Ren, AI-CoDesign Group Leader, BNL

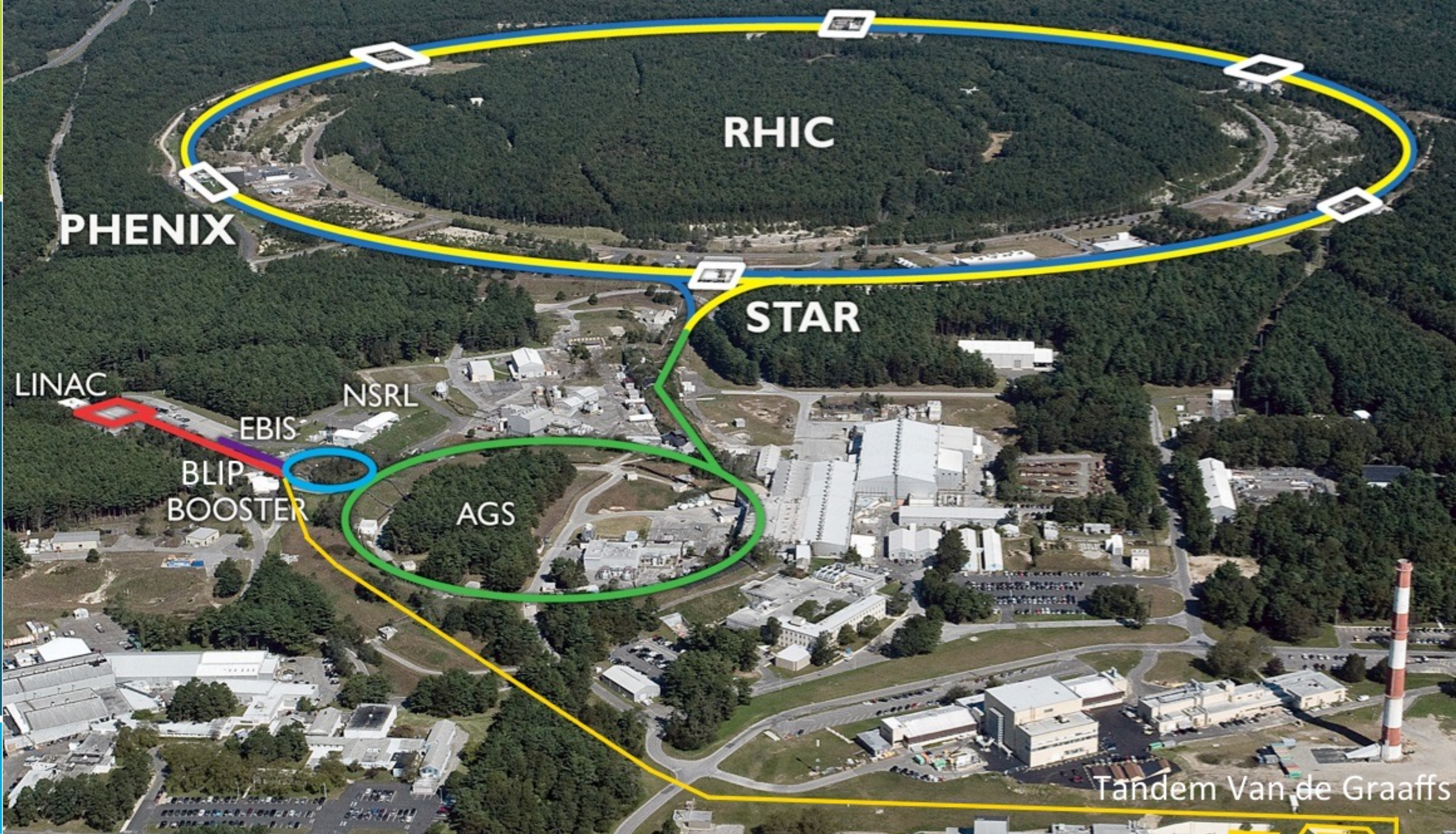
SRO XII, University of Tokyo, Dec. 2-4, 2024

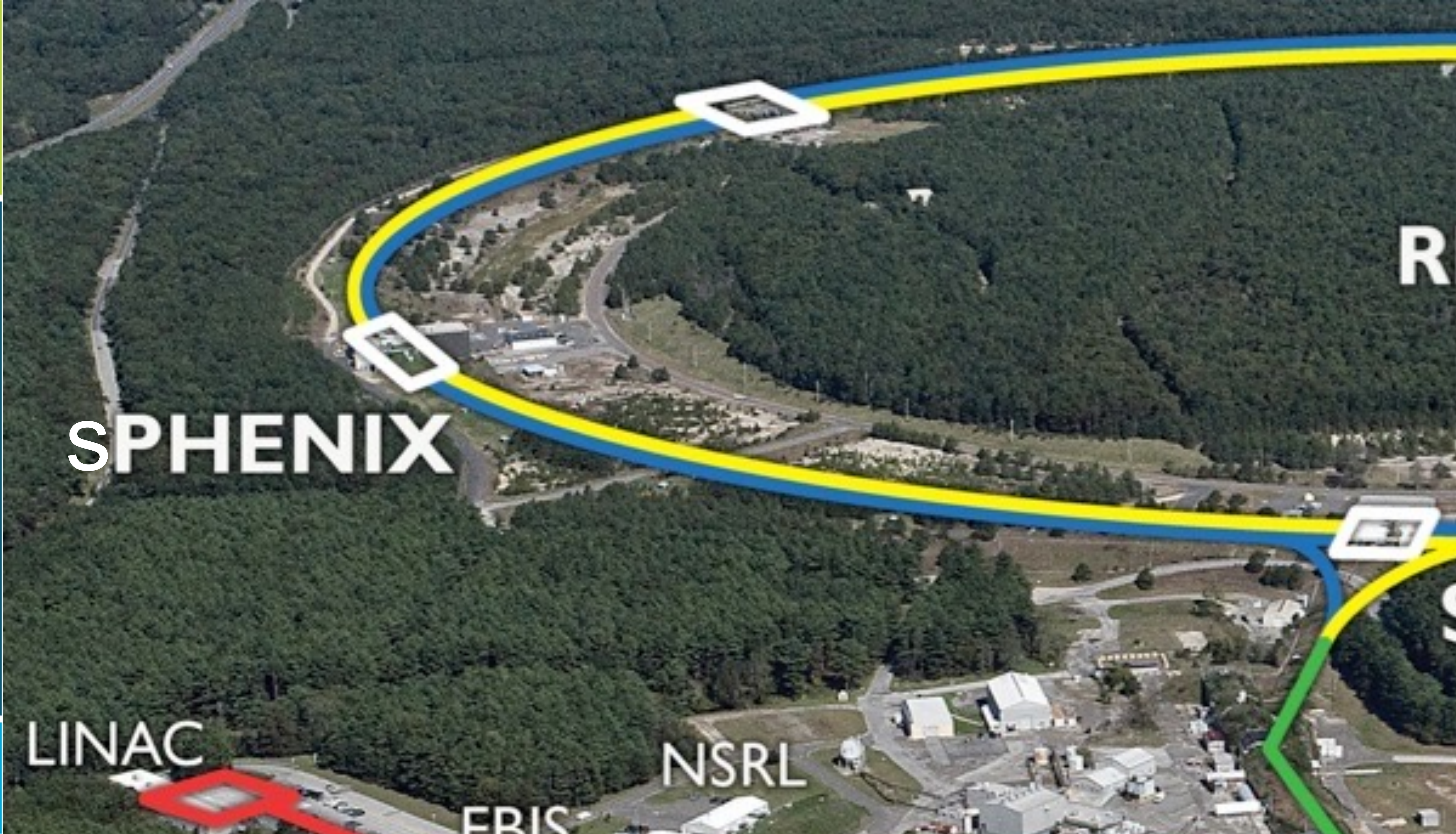

CT
NYC
Long Island
NJ
Atlantic Ocean



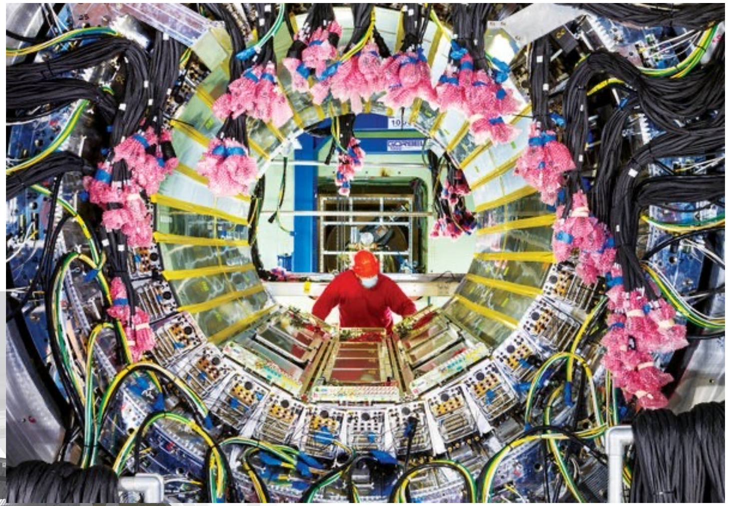
Relativistic Heavy Ion Collider,
future Electron-Ion Collider
(2.4 miles in circumference)
National Synchrotron
Light Source II


Prev: Computational
Science Initiative (CSI)
Now: Computing and Data
Science (CDS)

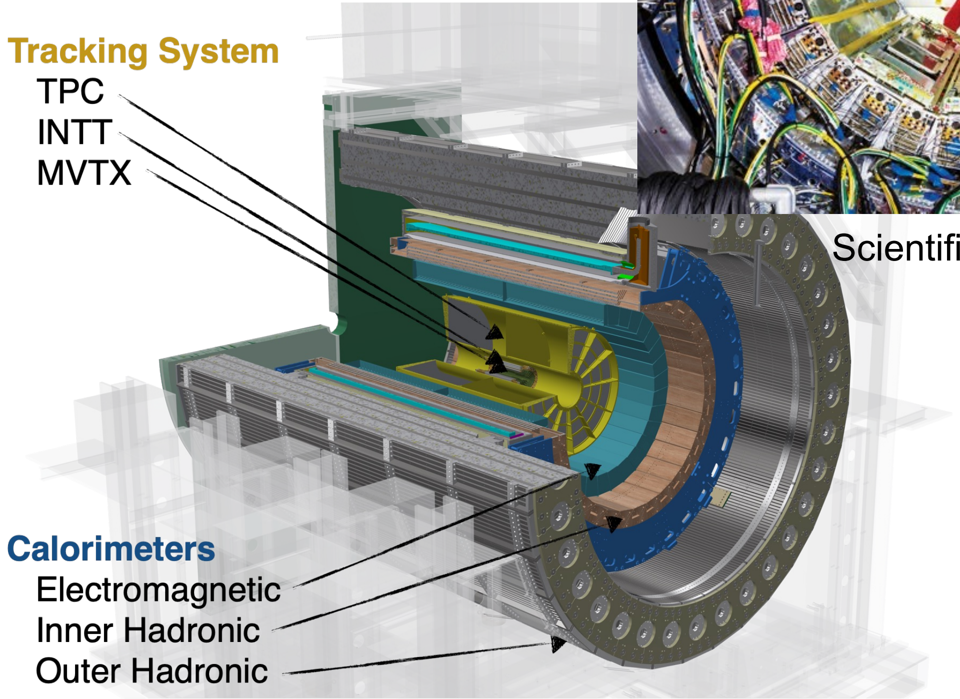

RHIC
PHENIX
STAR
LINAC
NSRL
EBIS
BLIP
BOOSTER
AGS
Tandem Van de Graaffs


SPHENIX
LINAC
NSRL

SPHENIX at BNL

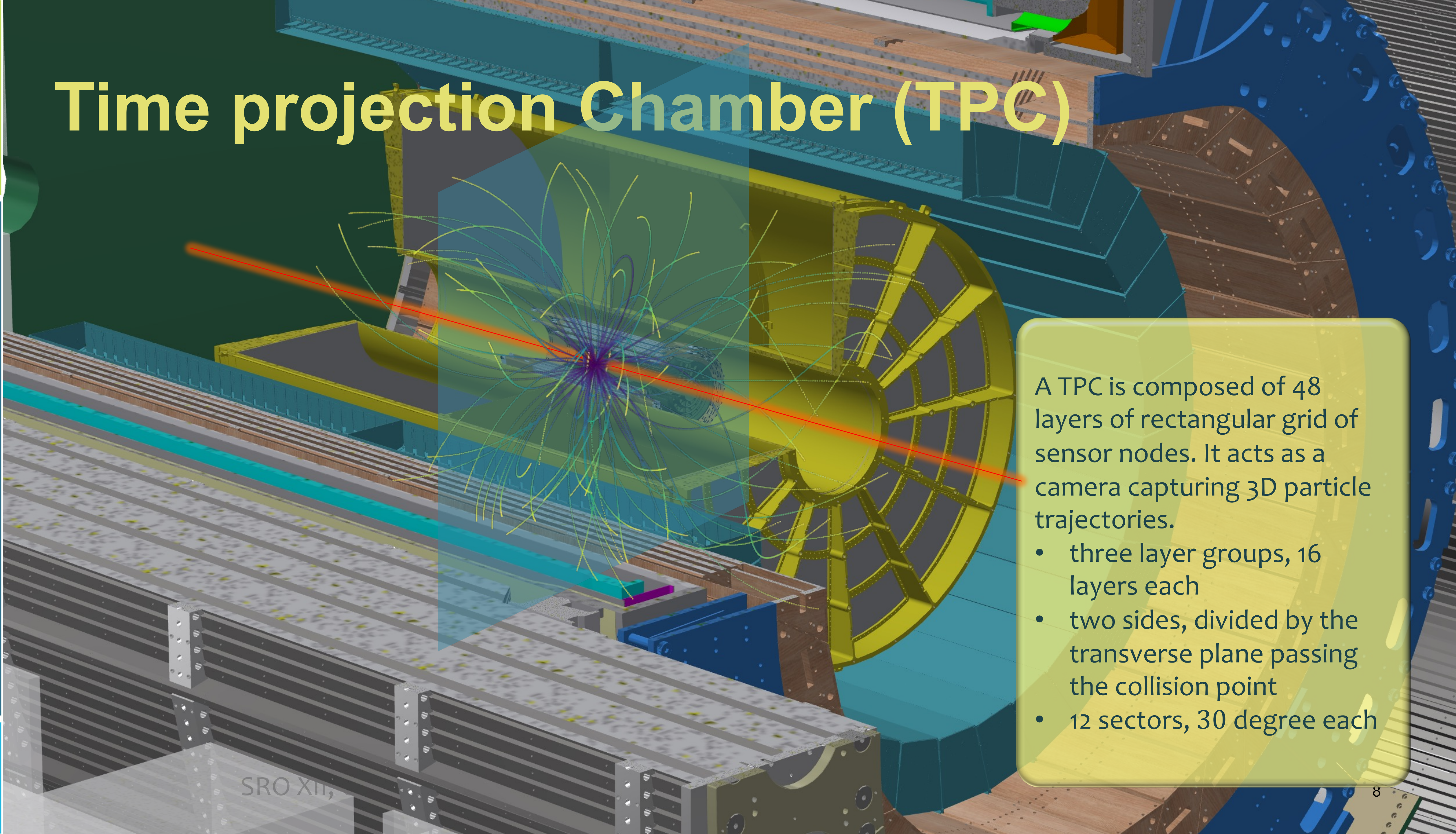


Scientific American, 03/01/2023



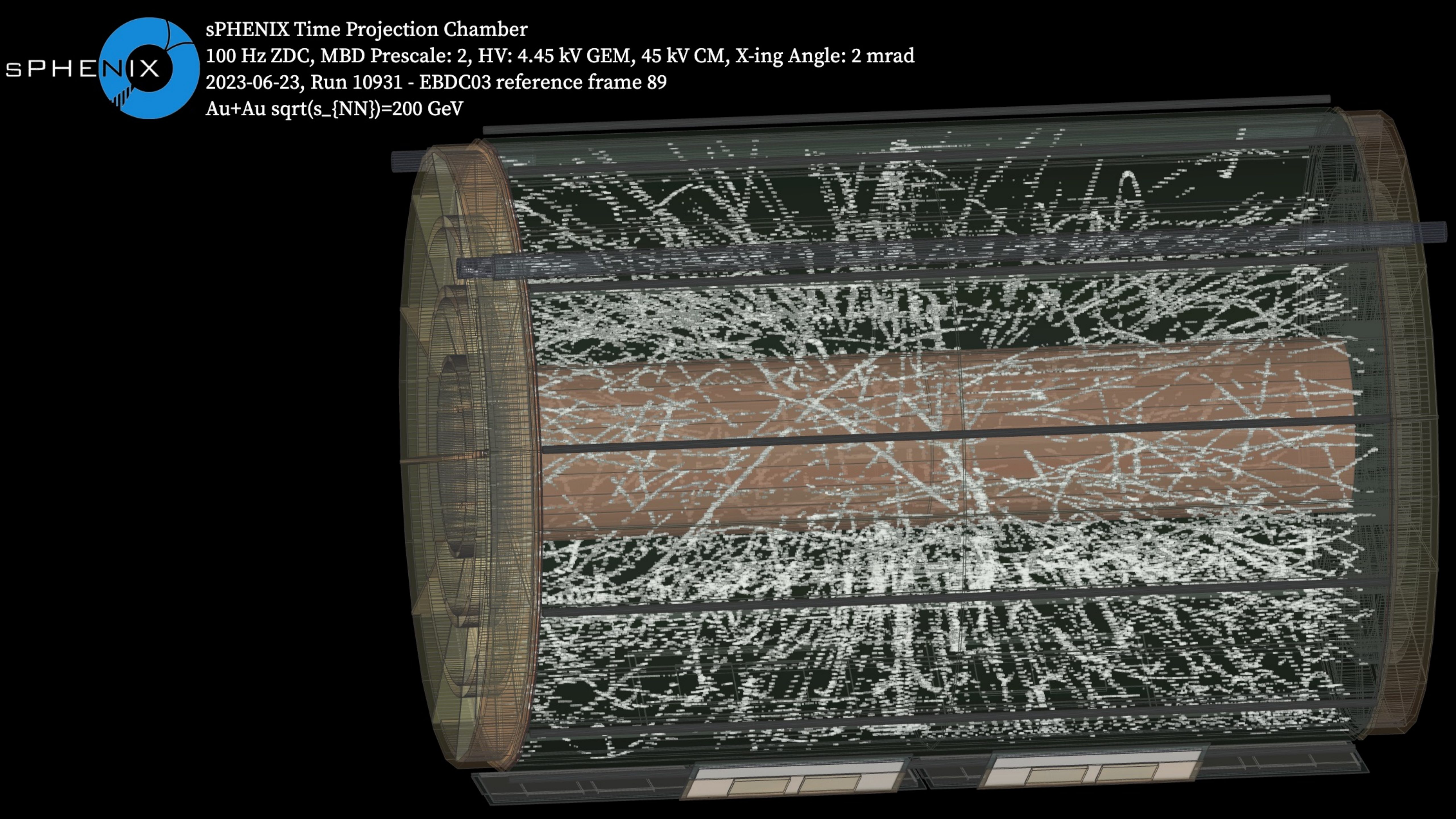
Data taking began last year!
High-precision tracking system + Hermetic Electromagnetic & Hadronic calorimeters

Time projection Chamber (TPC)



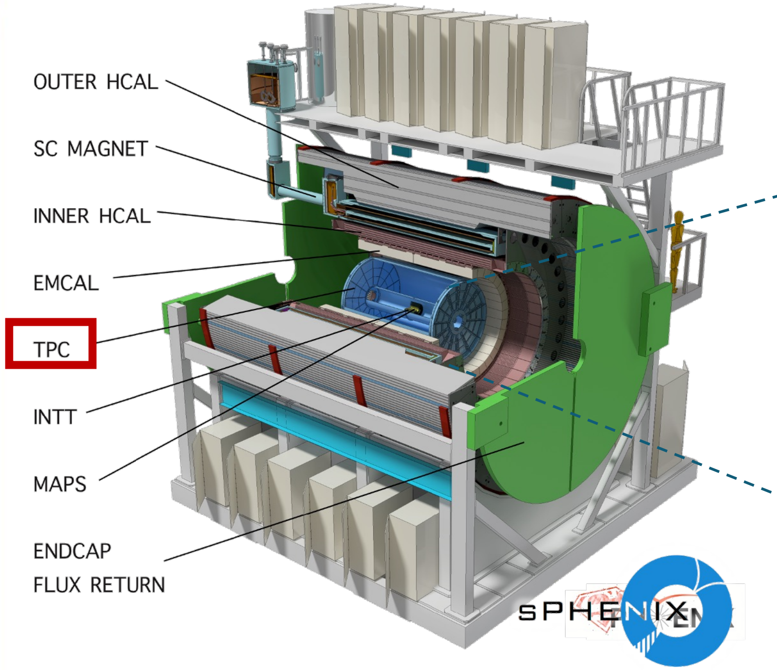
A TPC is composed of 48 layers of rectangular grid of sensor nodes. It acts as a camera capturing 3D particle trajectories.

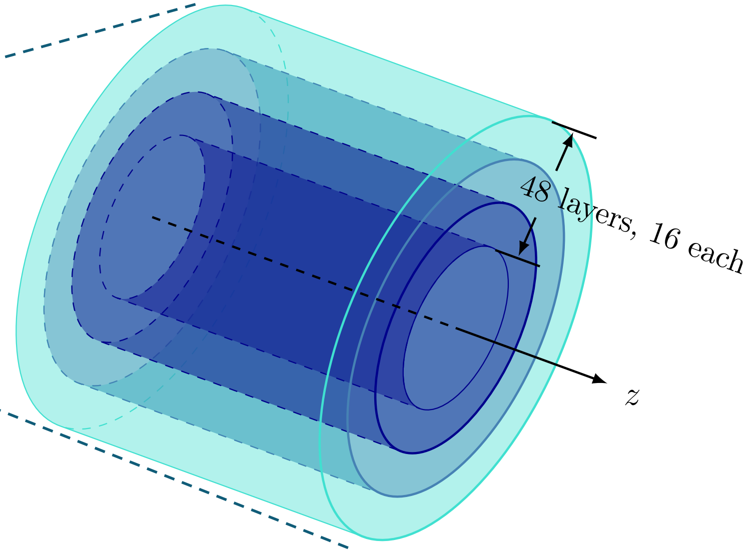
- three layer groups, 16 layers each
- two sides, divided by the transverse plane passing the collision point
- 12 sectors, 30 degree each

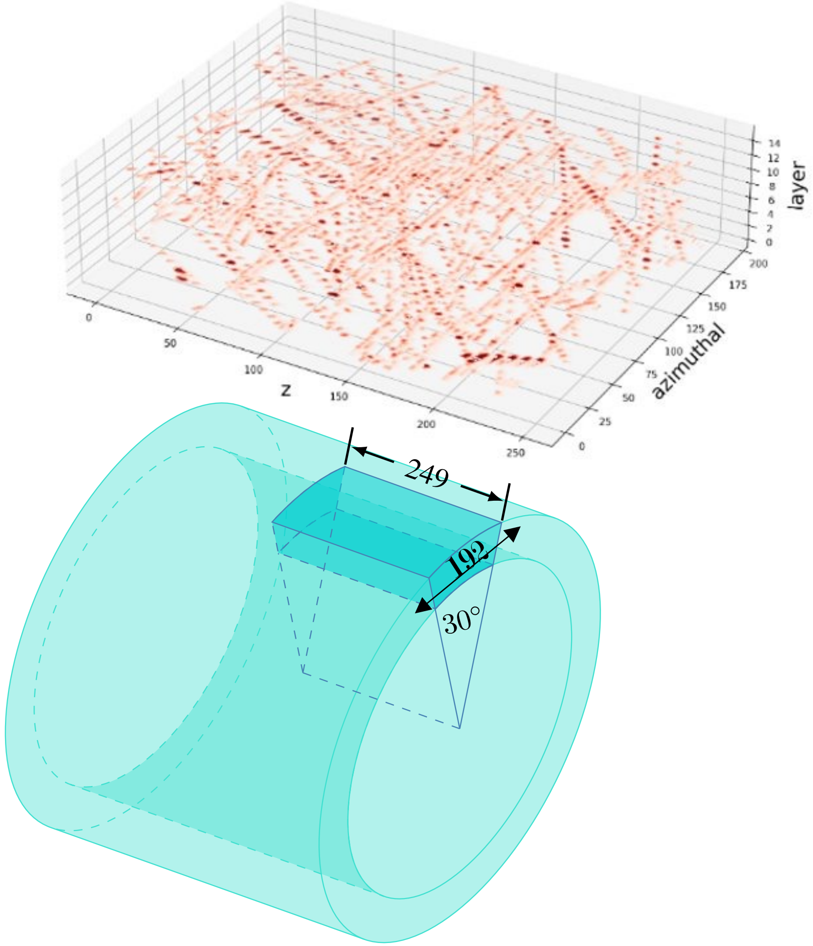

sPHENIX
sPHENIX Time Projection Chamber
100 Hz ZDC, MBD Prescale: 2, HV: 4.45 kV GEM, 45 kV CM, X-ing Angle: 2 mrad
2023-06-23, Run 10931 - EBDC03 reference frame 89
Au+Au sqrt(s_{NN})=200 GeV

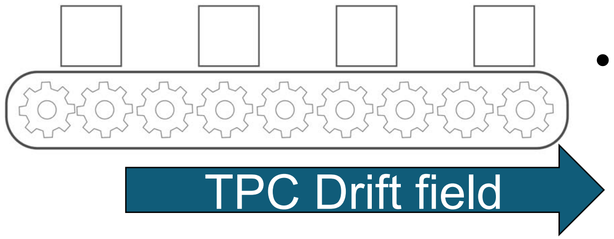
TPC Data

Dataset: MDC1 AuAu 0-10%C + 170kHz pileup







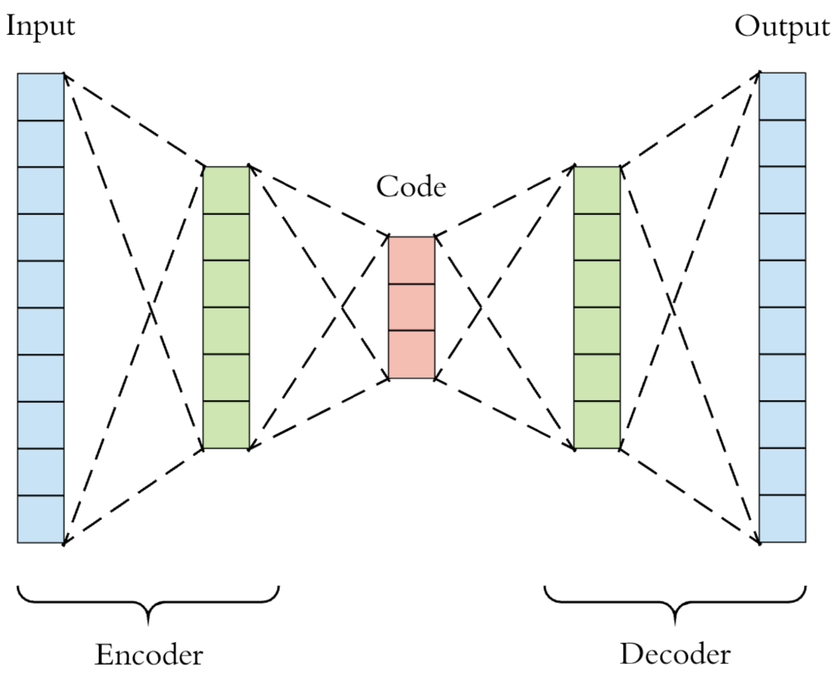


- Number of voxels: (azimuth × z × layer):
  - Outer layer group: 2304 × 498 × 16 ≈ 18M;
  - Middle layer group: 1536 × 498 × 16 ≈ 12M;
  - Inner layer group: 1152 × 498 × 16 ≈ 9M

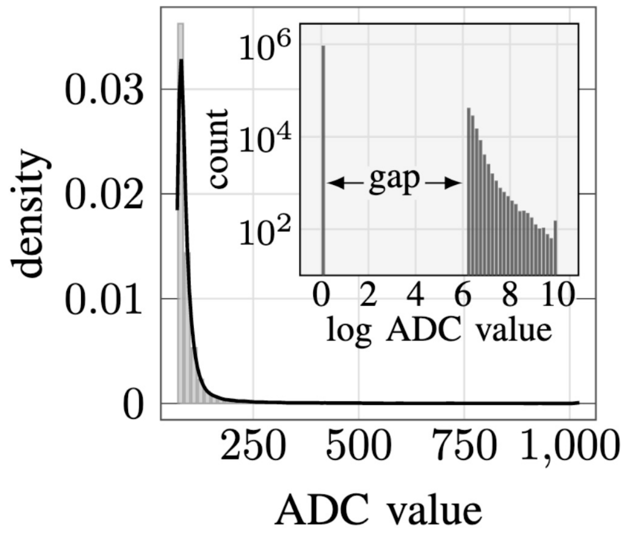
Outer Layer Section:
- 192 × 249 × 16
- Sparser

Neural Auto-Encoder

- A typical Auto-Encoder uses an Encoder network to compress the data into “code”; and a Decoder network to reconstruct the original input.



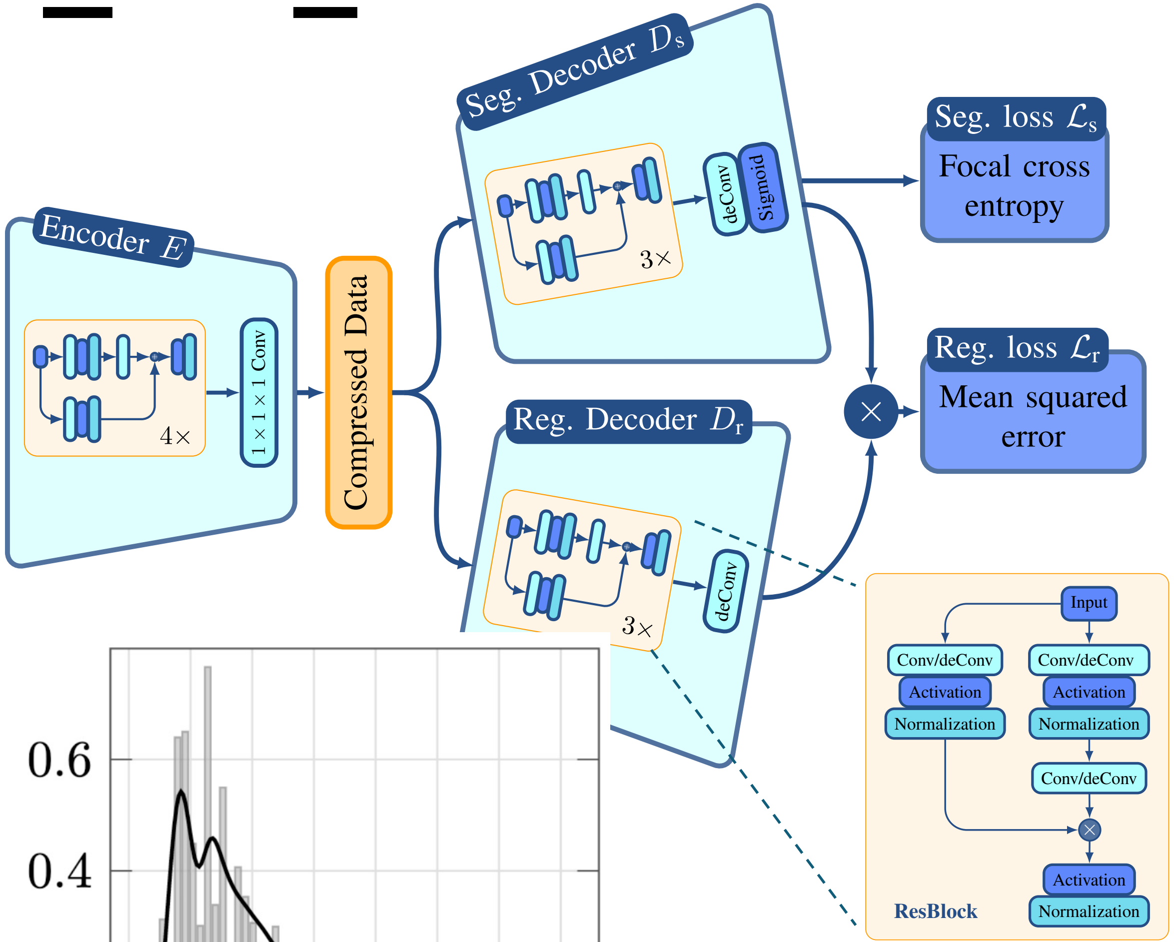
- The voxel distribution:
  - long-tailed (skewed)
  - sparse (many zero values)
  - zero-suppressed (discontinued)
  - 10-bit integer (saturated)

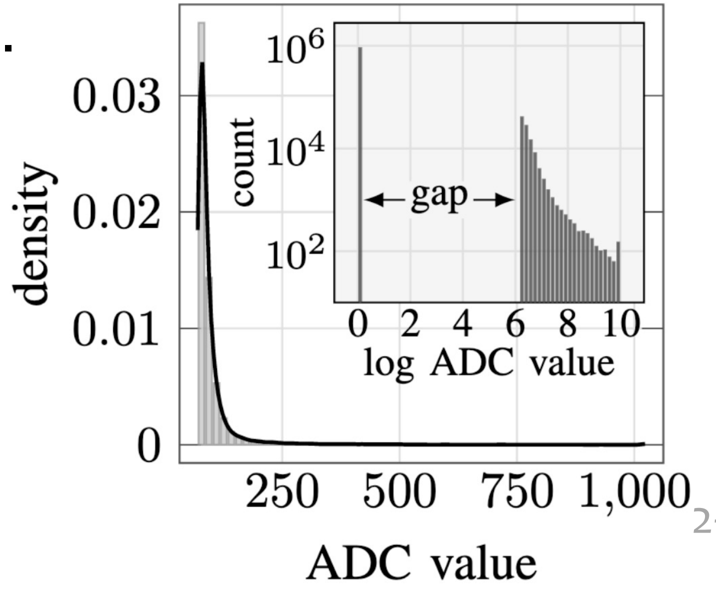


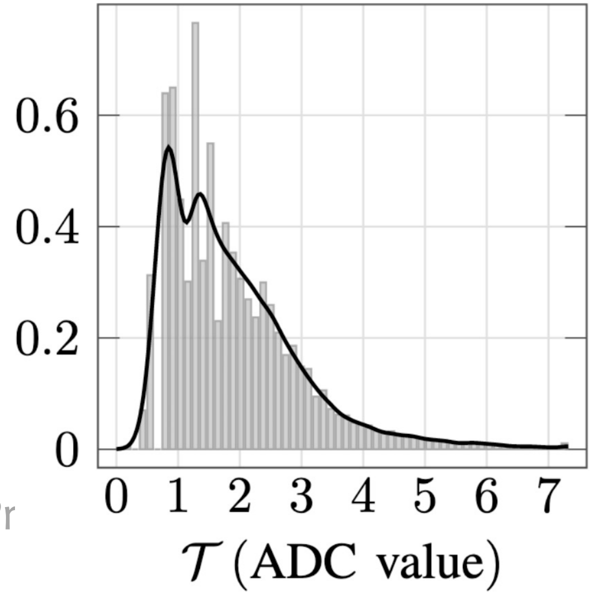
Very Challenging for a regular auto-encoder!

<u>B</u>icephalous <u>C</u>onvolutional <u>A</u>uto-<u>E</u>ncoder (BCAE)

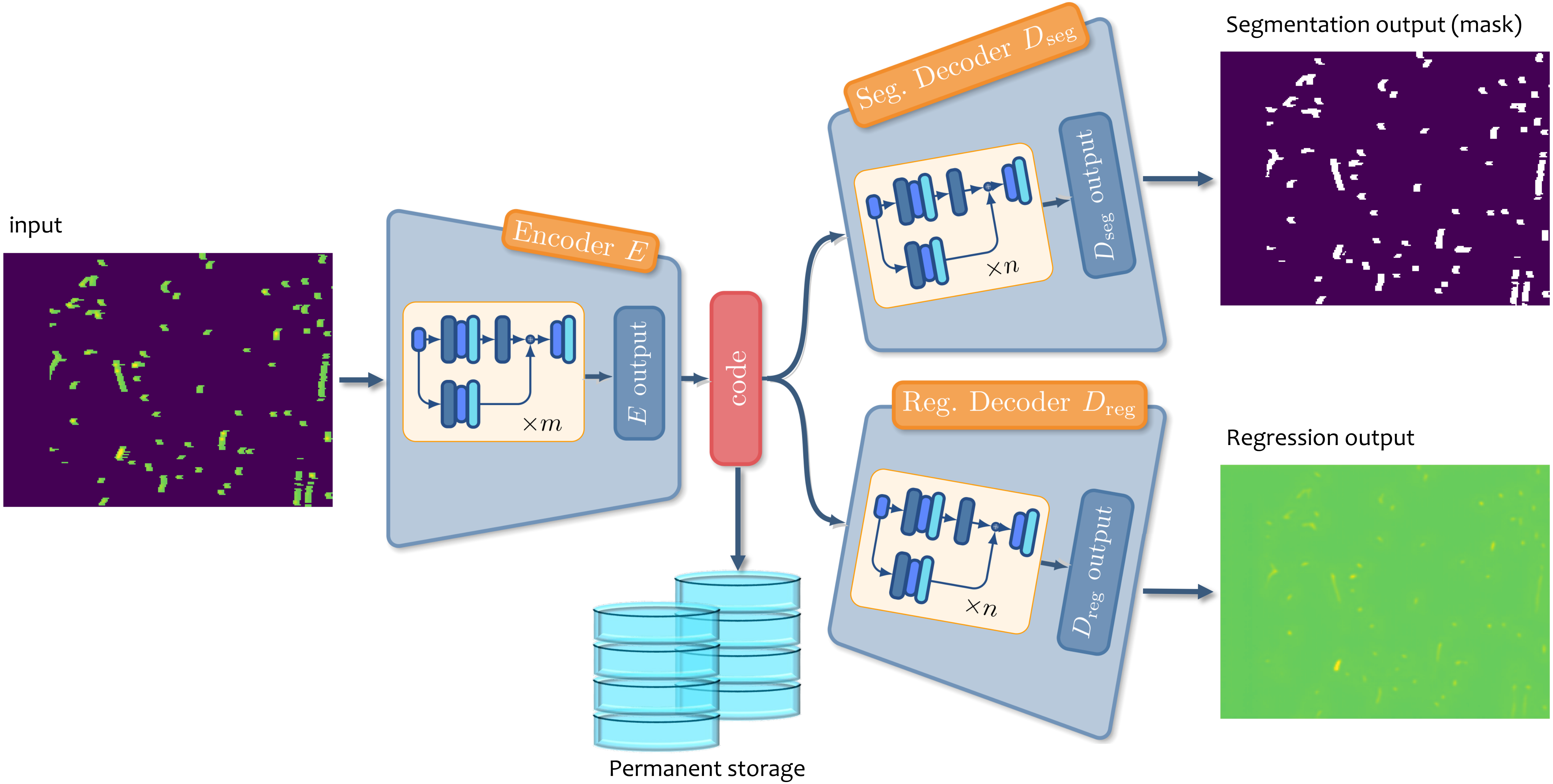
- A dedicated Segmentation decoder to determine whether a voxel has been zero-suppressed.
- Integrate a transformation function τ into the network: $\tau = \log(x - 64)/6$ for non-zero values.



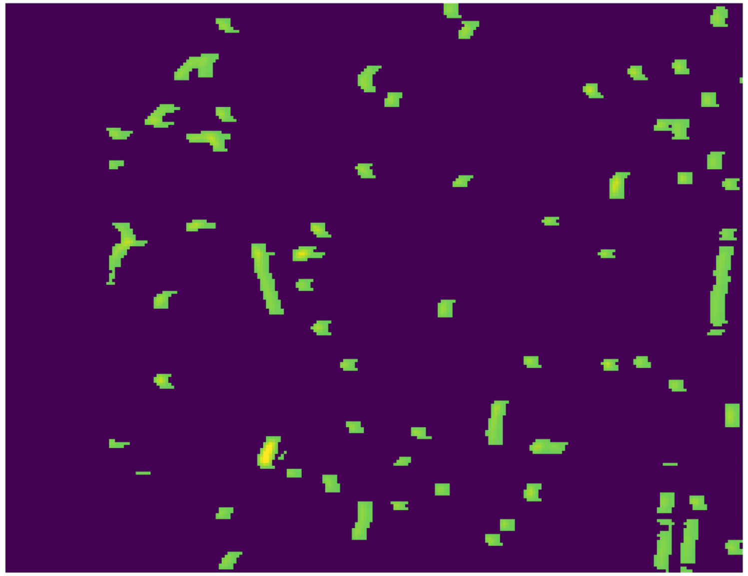




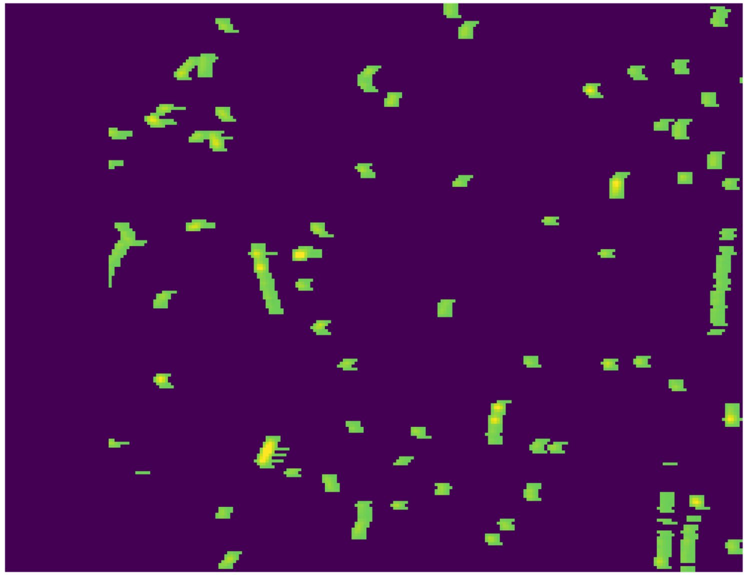
2-4, 2024. Pr


input
Encoder E
$\times m$
E output
code
Seg. Decoder D_{seg}
$\times n$
D_{seg} output
Reg. Decoder D_{reg}
$\times n$
D_{reg} output
Segmentation output (mask)
Regression output
Permanent storage

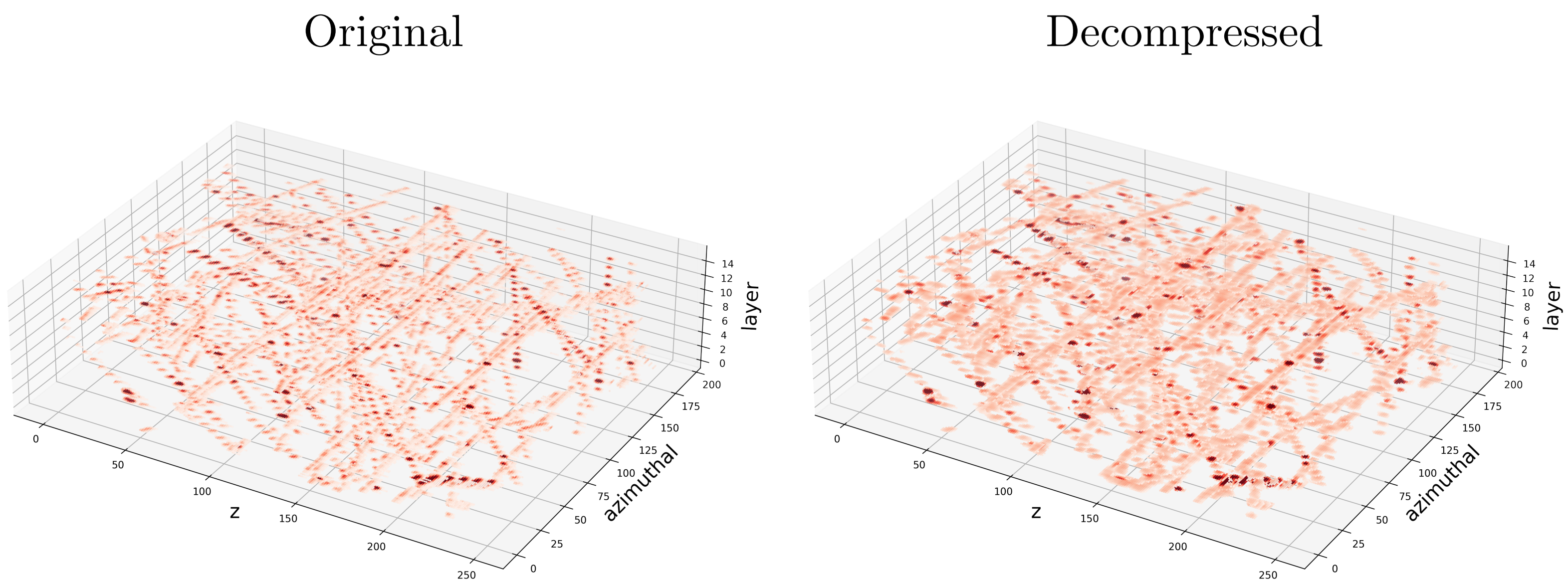
Regression output × mask



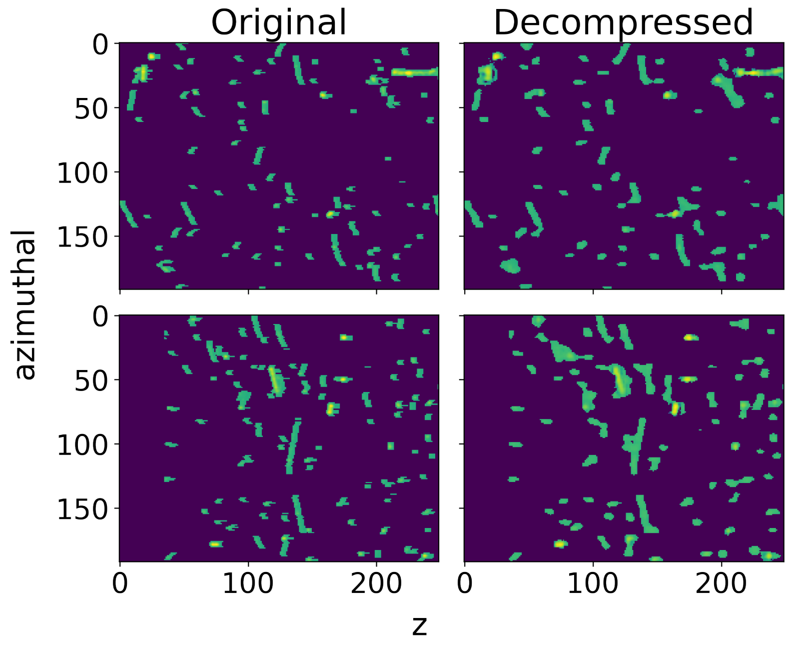
input



Results



- Compression Ratio: **1:27**
- Mean-Squared Error (MSE): **218.44**



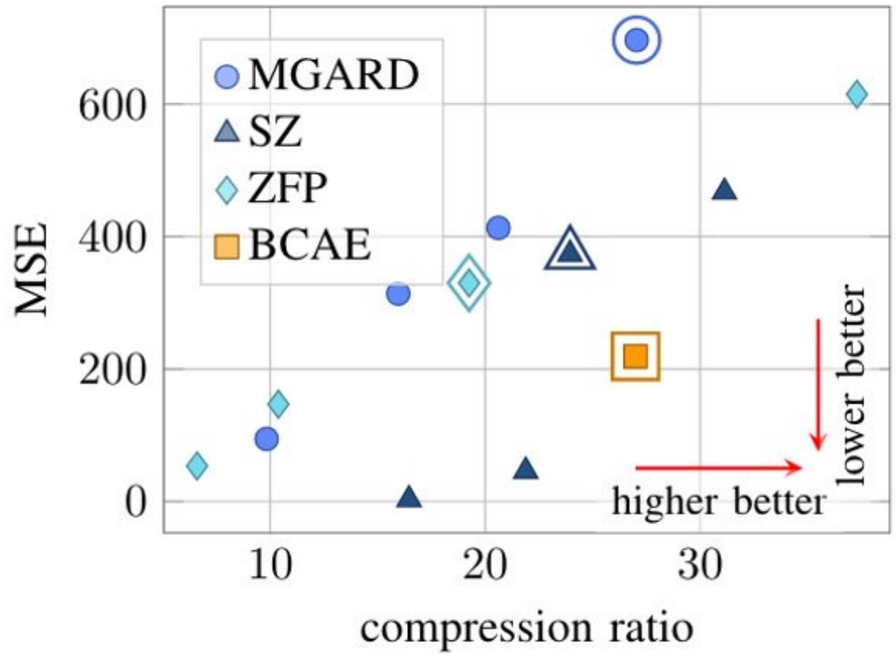
Other Lossy Compression Algorithms

We identified three conventional lossy compression algorithms, which were mainly designed for dense data matrices such as in fluid dynamic simulations.

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

Results

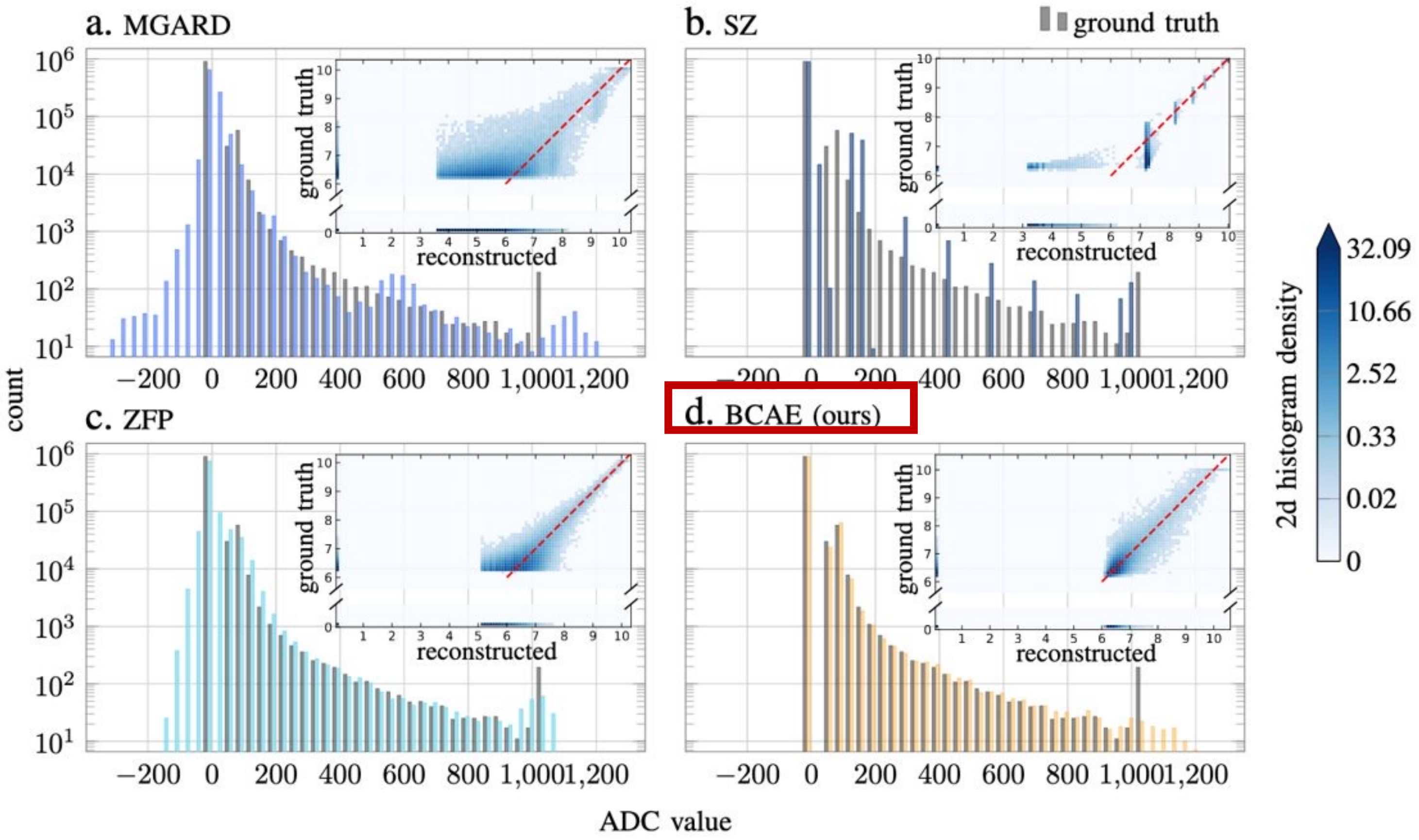
- Conventional methods allow users to change compression ratio.
- Our model has better compression ratio and lower MSE. (good balance)
- Conventional methods do not require “training”.



	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
CAE	27	227.61	0.349	3.703
BCAEwoT	27	230.59	0.193	3.706
BCAE	**27**	**218.44**	**0.185**	**3.724**

Results

- Conventional methods allow users to change compression ratio.
- Our model has better compression ratio and lower MSE.
- Recover entry value distributions (histogram)

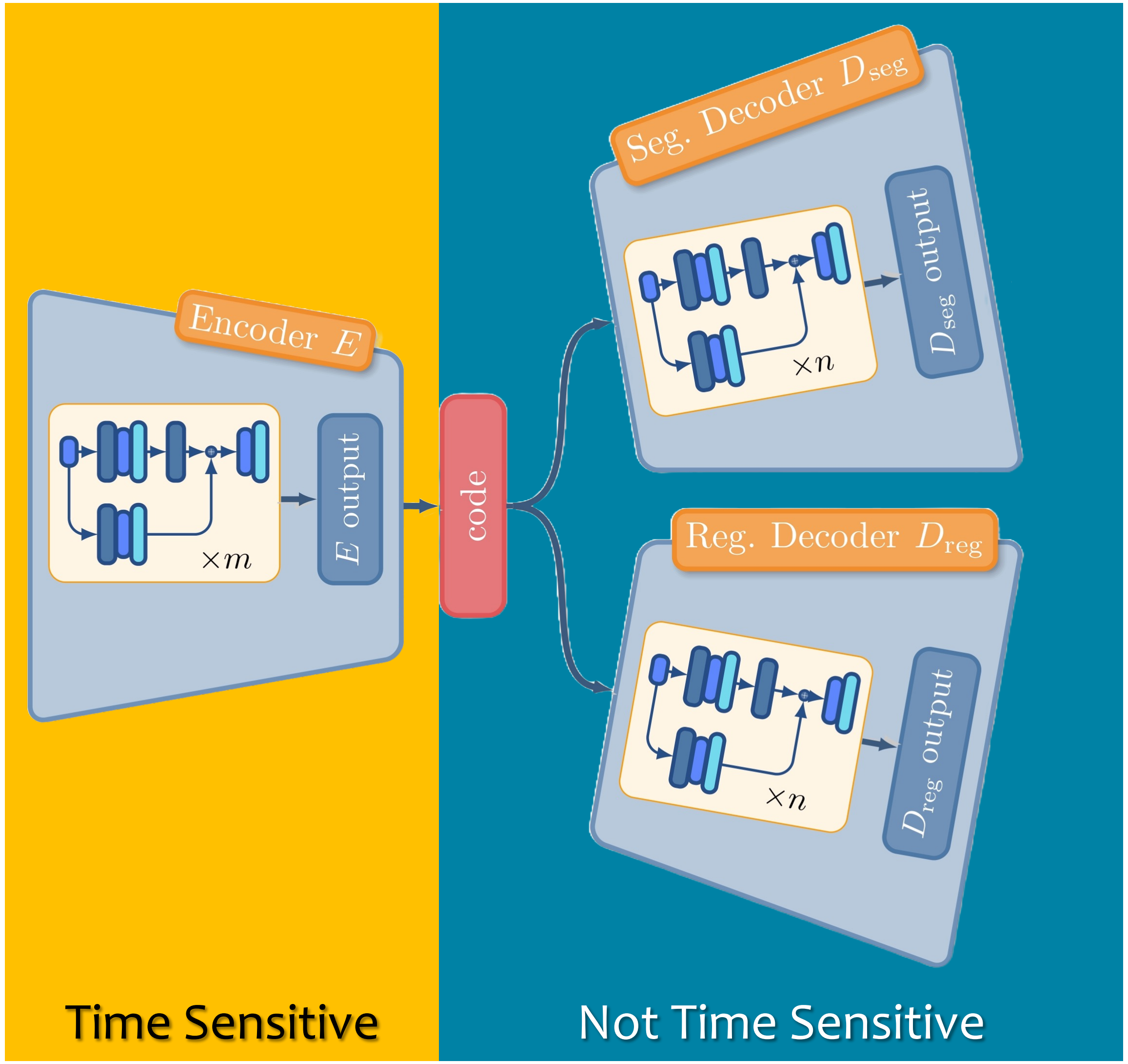


Huang, Y., Ren, Y., Yoo, S., & Huang, J. (2021, December). Efficient data compression for 3d sparse TPC via bicephalous convolutional autoencoder. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 1094-1099). IEEE. arxiv:2111.05423

Jin Huang (BNL) “SRO for sPHENIX TPC and Real-time AI” in SRO XI

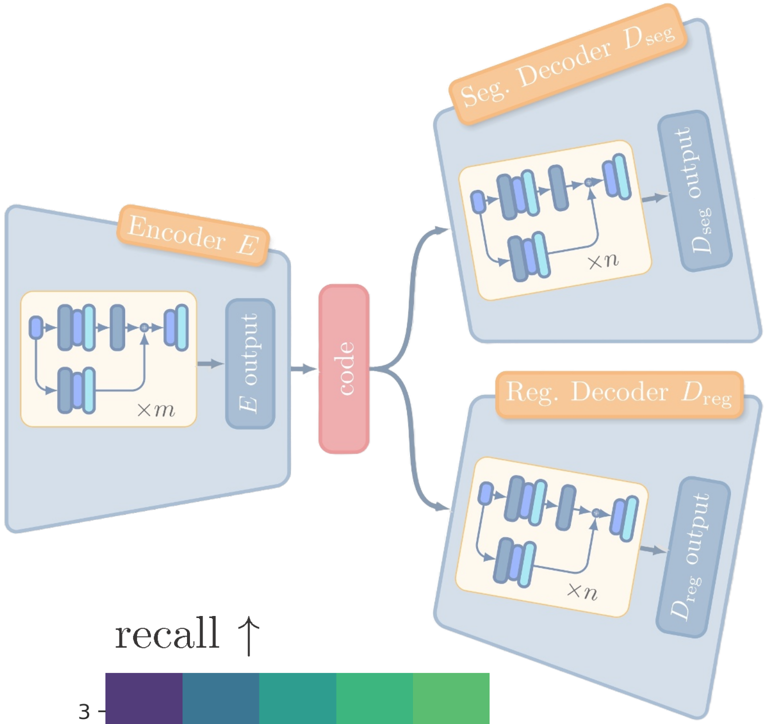
2D Encoder and Decoder

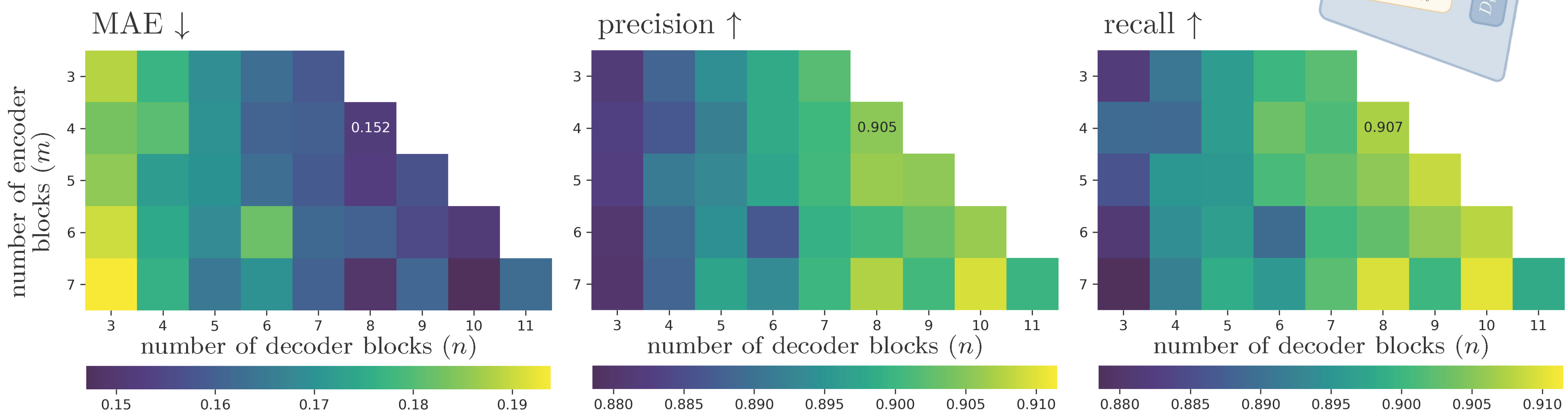
- Smaller, faster encoder
- Bulkier, slower decoder
- Can stronger decoder compensate for a weaker encoder?



2D Encoder and Decoder

tunable encoder decoder sizes





Performance comparison

original BCAE versus BCAE-2D

model	MAE ↓	PSNR ↑	Precision ↑	Recall ↑	Encoder size ↓	Compr. Ratio ↓
BCAE	0.198	9.923	0.878	0.861	201.7k	27.041
BCAE-2D	0.152	11.726	0.906	0.907	169.0k	31.125

Performance comparison

original BCAE versus BCAE-2D

model	MAE ↓	PSNR ↑	Precision ↑	Recall ↑	Encoder size ↓	Compr. Ratio ↓
BCAE	0.198	9.923	0.878	0.861	201.7k	27.041
BCAE-2D	0.152	11.726	0.906	0.907	169.0k	31.125
BCAE++	0.112	14.325	0.934	0.936	226.2k	31.125

From BCAE to BCAE++

1. 3D convolution
2. Pad (16, 192, 249) to (16, 192, 256) for easy halving and an increased compression ratio
3. Remove normalization

- Better reconstruction performance
- Still slow

Performance comparison

original BCAE versus BCAE-2D

model	MAE ↓	PSNR ↑	Precision ↑	Recall ↑	Encoder size ↓	Compr. Ratio ↓
BCAE	0.198	9.923	0.878	0.861	201.7k	27.041
BCAE-2D	0.152	11.726	0.906	0.907	169.0k	31.125
BCAE++	0.112	14.325	0.934	0.936	226.2k	31.125
BCAE-HT	0.138	12.376	0.916	0.915	$\boxed{9.8k}$	31.125

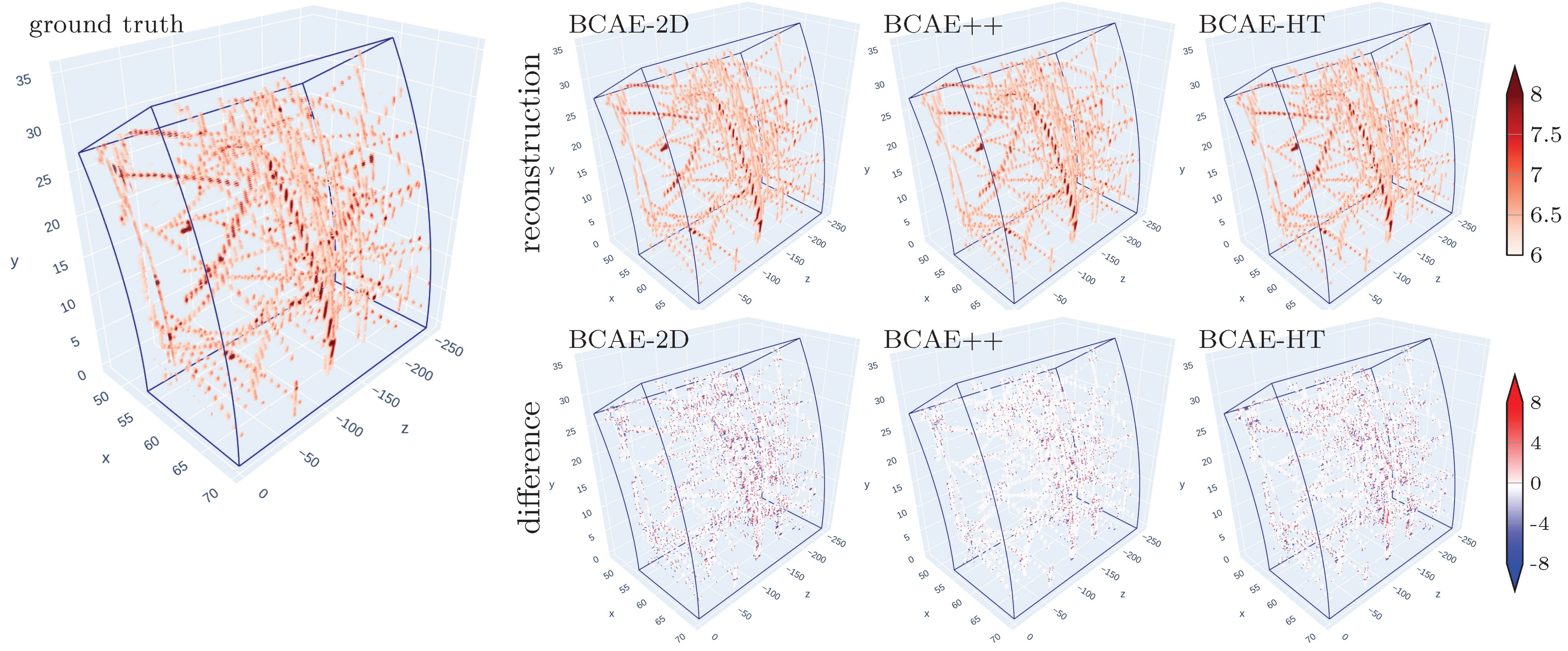
From BCAE to BCAE-HT

1. 3D convolution
2. Pad (16, 192, 249) to (16, 192, 256) for easy halving and an increased compression ratio
3. Remove normalization
4. Much smaller intermediate output channels for higher throughput

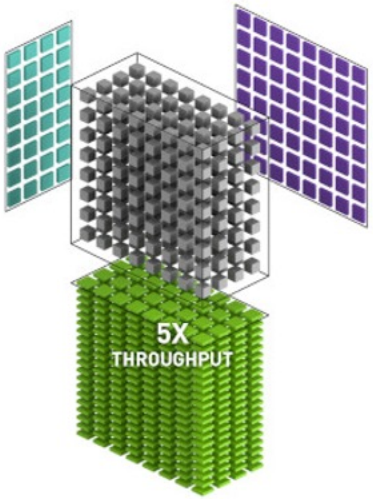
- Slightly better reconstruction performance
- Super small model size
- Higher throughput

Performance comparison

original BCAE versus BCAE-2D



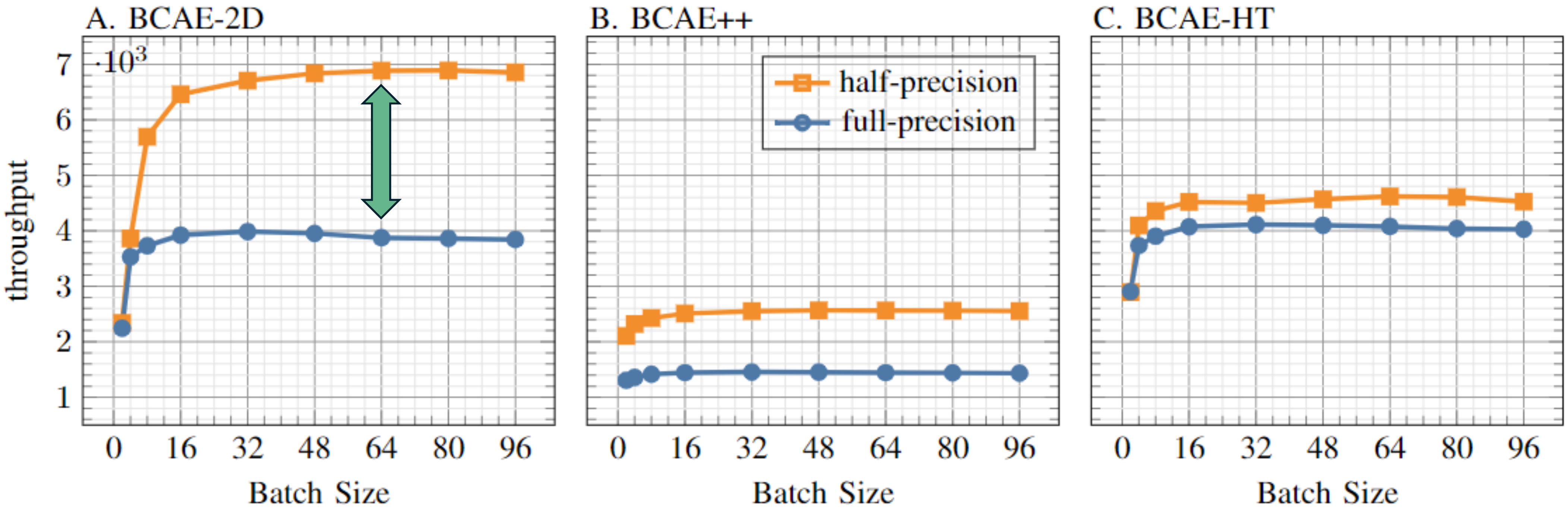
Throughput comparison



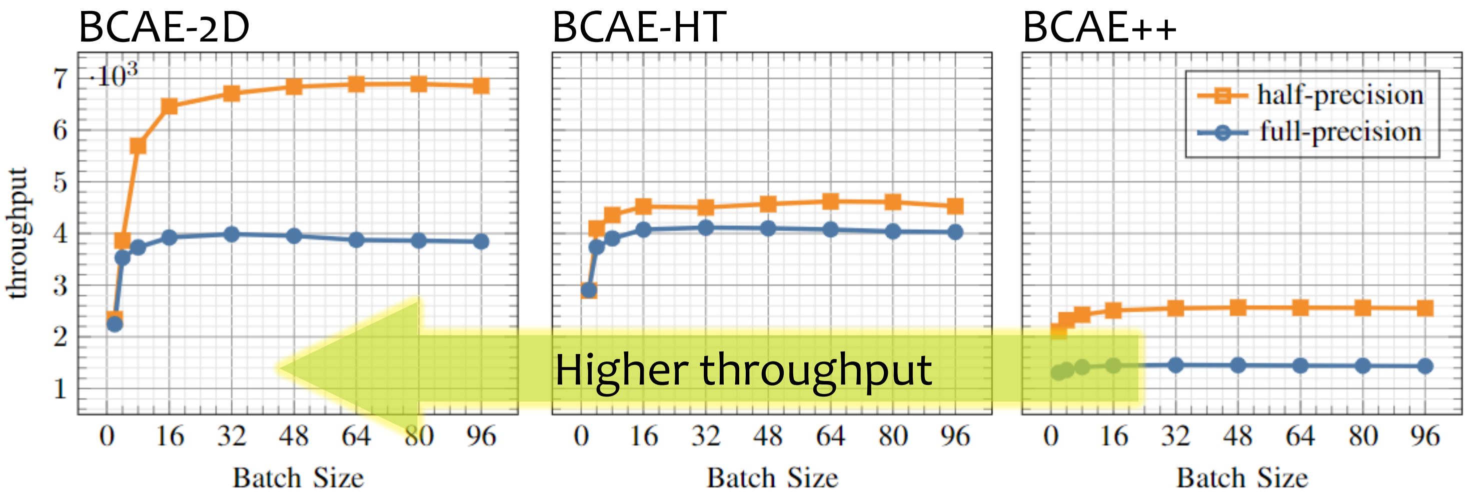
- **Full:** encode with float 32, save code as float 16, decode with float 32
- **Half:** encode with float 16, save code as float 16, decode with float 32

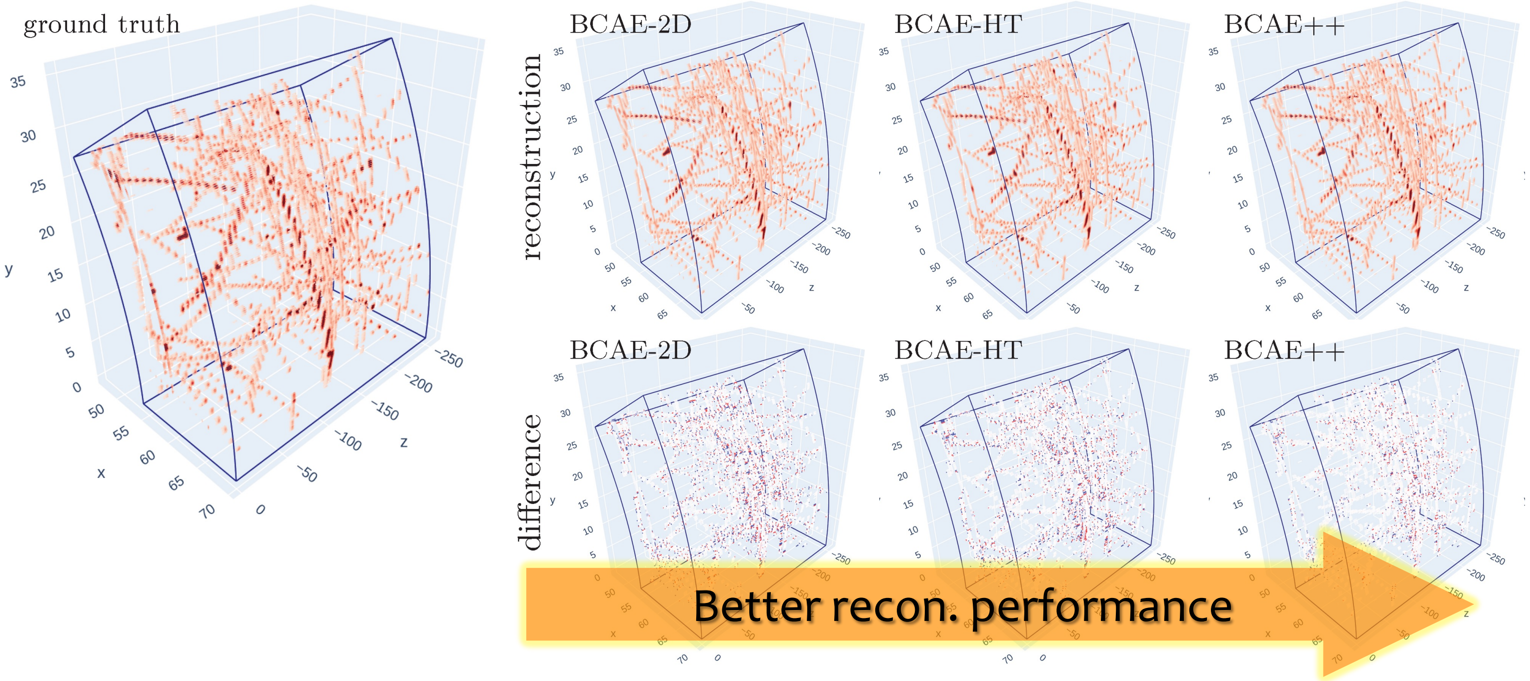
model	mode	MAE	precision	recall
BCAE-2D	Full	0.151937	0.905469	0.906916
	Half	0.151965	0.905326	0.907050
BCAE++	Full	0.112347	0.933817	0.935779
	Half	0.112342	0.933852	0.935741
BCAE-HT	Full	0.138443	0.915891	0.914562
	Half	0.138441	0.915780	0.914701

Throughput comparison



Measured on A6000





Huang, Y., Ren, Y., Yoo, S., & Huang, J. (2023, November). Fast 2D Bicephalous Convolutional Autoencoder for Compressing 3D Time Projection Chamber Data. In *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis* arxiv:2310.15026

Question 1:

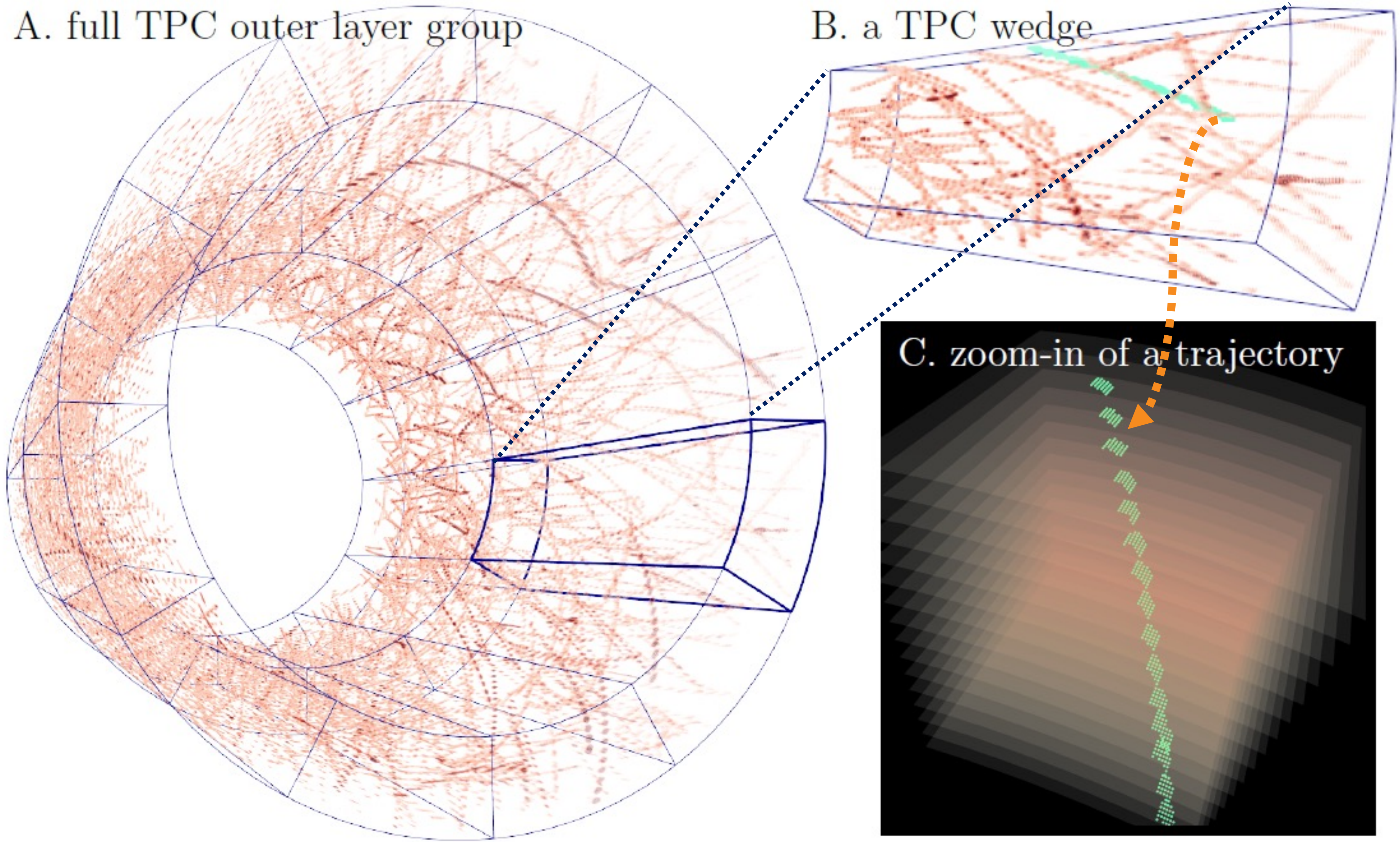
Can we have better performance and better throughput?

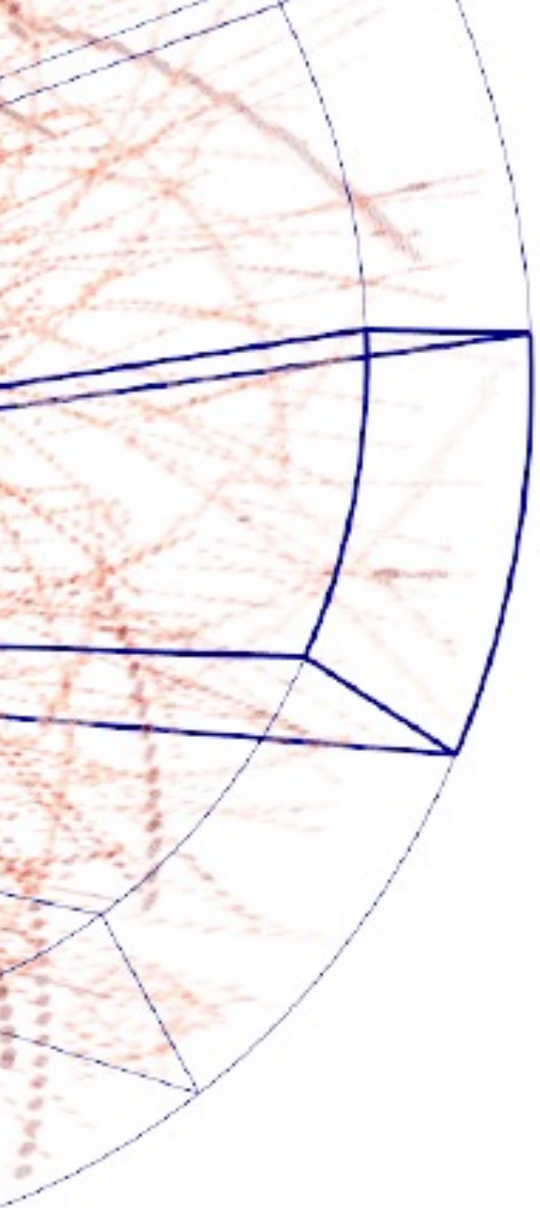
Question 2:

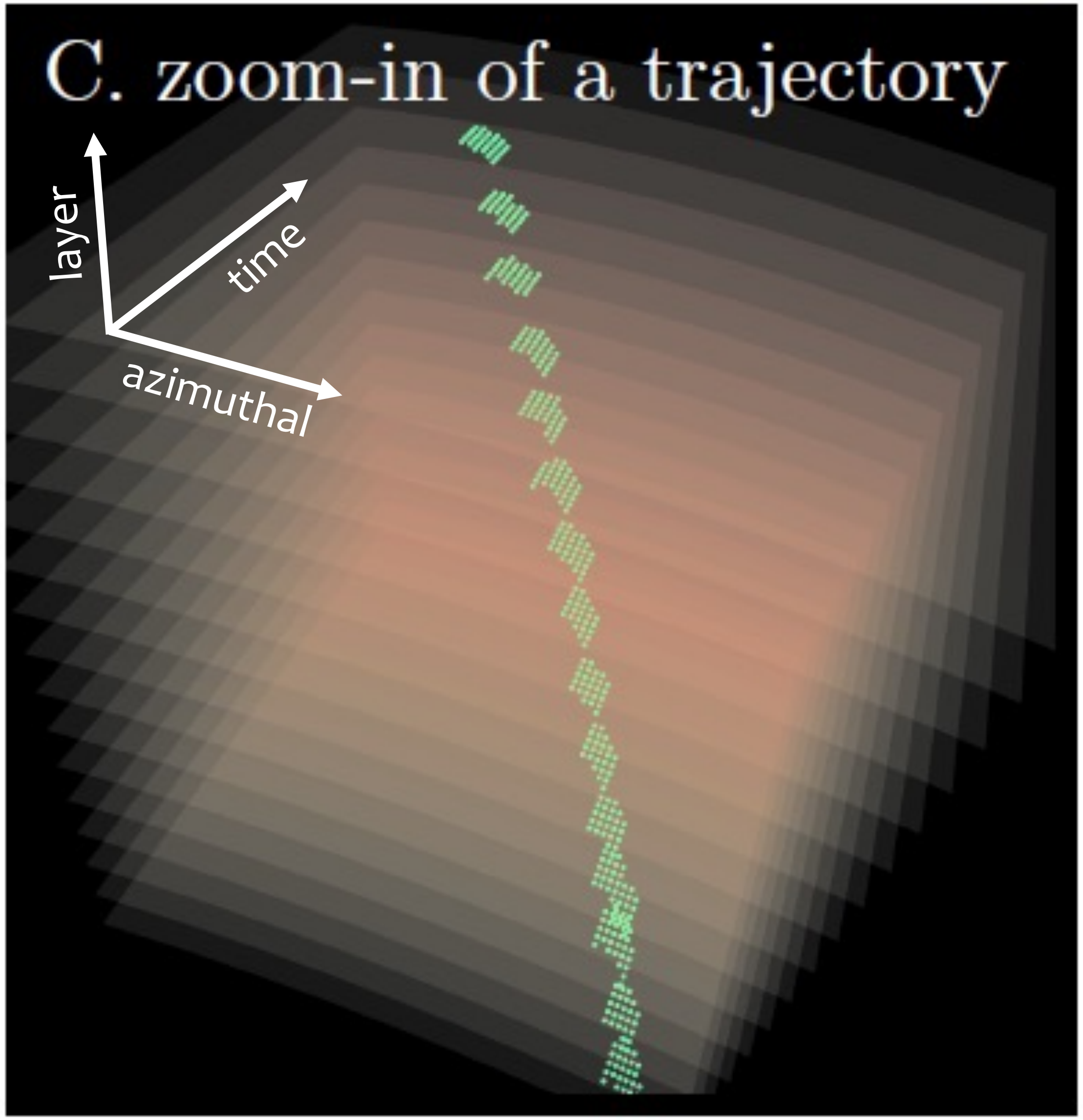
Can we have variable compression ratio depending on occupancy?

Question 3:

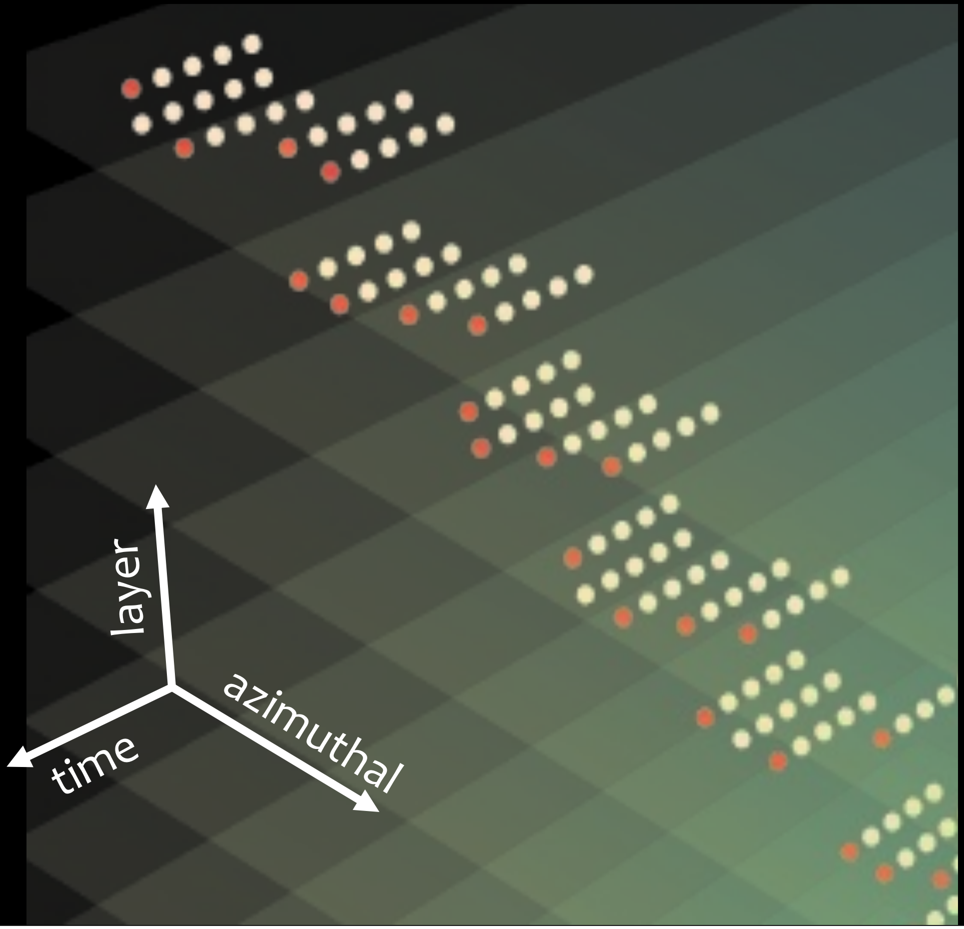
Can we have variable throughput? Sparser the data, the less compute


A. full TPC outer layer group
B. a TPC wedge
C. zoom-in of a trajectory




C. zoom-in of a trajectory
layer
time
azimuthal

BCAE-VS: Bicephalous Convolutional Autoencoder with Variable ratio Compression for Sparse input



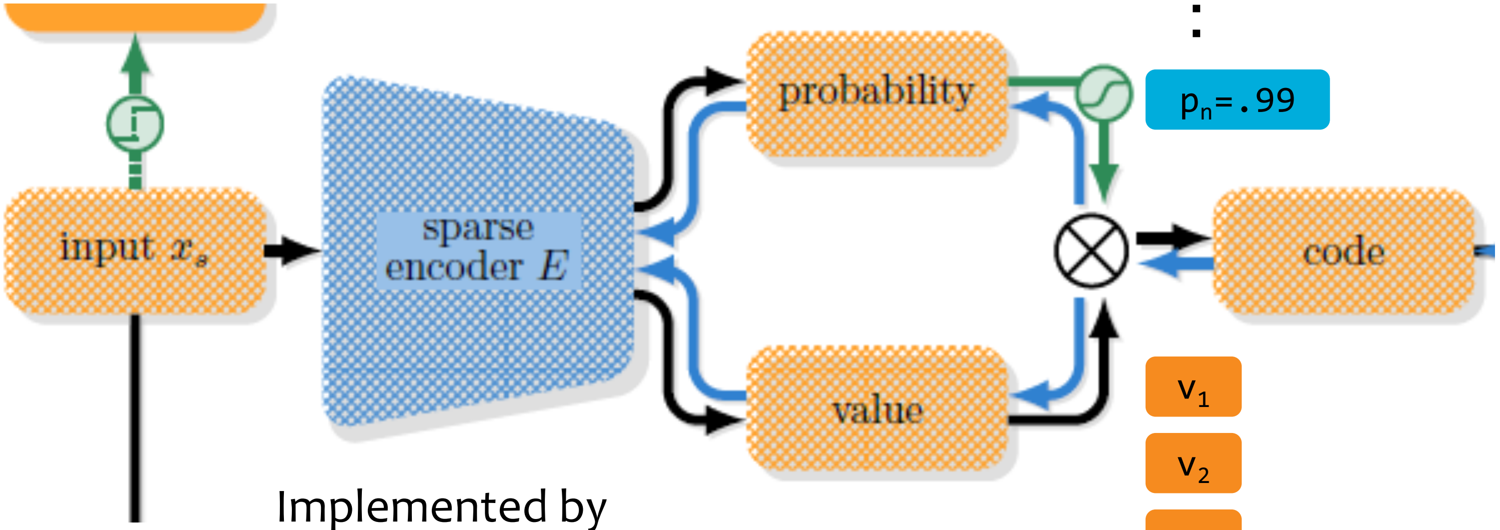
Locate the most valuable signals, and compress by down-selecting the signals

BCAE-VS

Bicephalous Convolutional Autoencoder with Variable ratio Compression for Sparse input

Sparse input

x_1, y_1, z_1, a_1
x_2, y_2, z_2, a_2
x_3, y_3, z_3, a_3
⋮
x_n, y_n, z_n, a_n



p_1=.92
p_2=.03
p_3=.12
⋮
p_n=.99

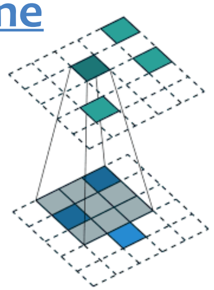
Threshold = .9

v_1
v_2
v_3
⋮
v_n

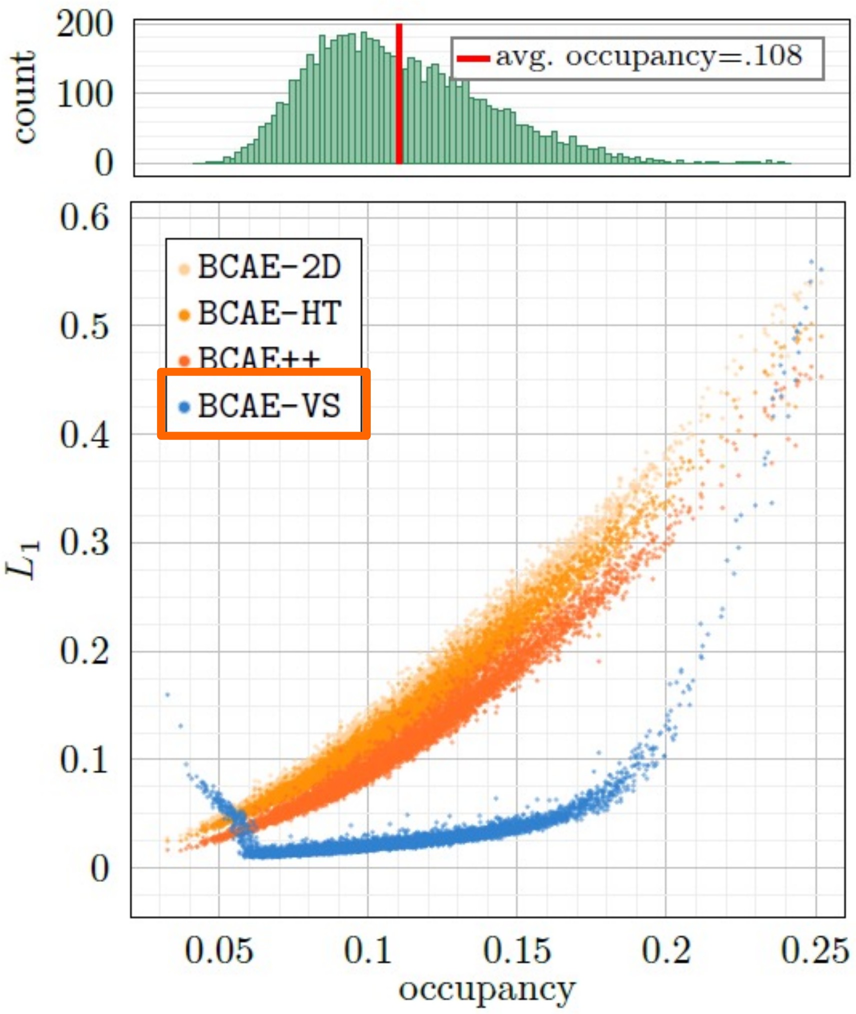
Sparse code

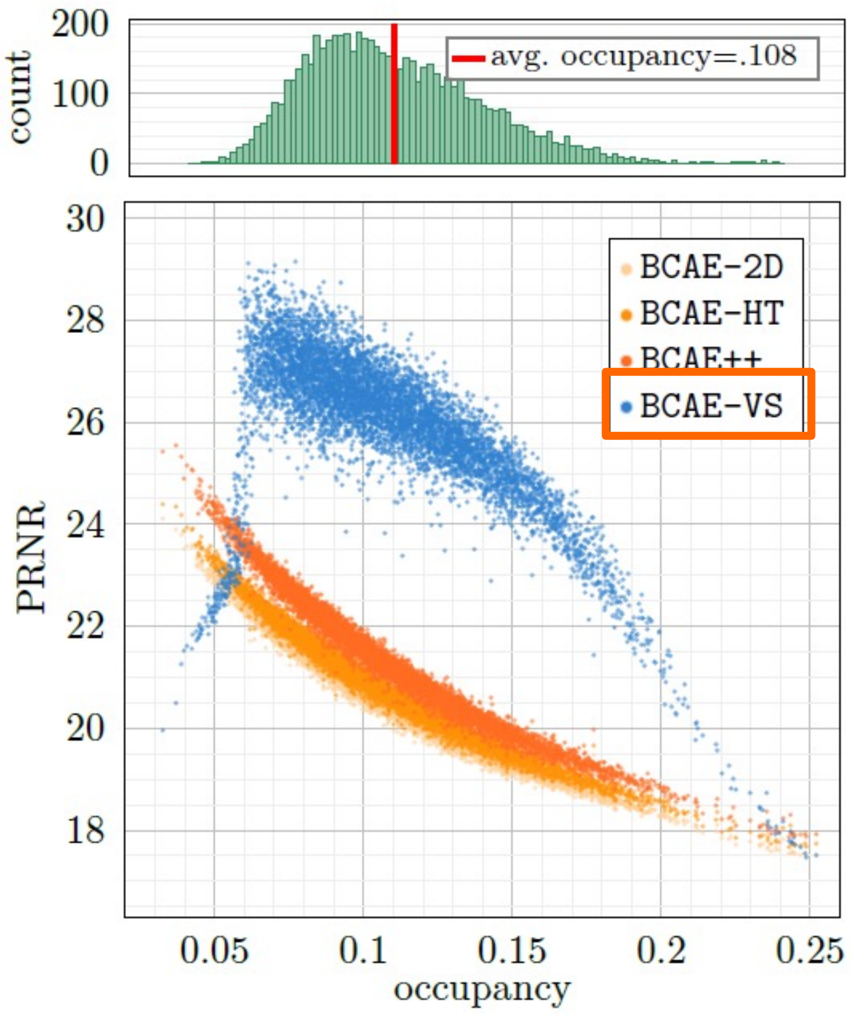
x_1, y_1, z_1, v_1
x_2, y_2, z_2, v_2
x_3, y_3, z_3, v_3
⋮
x_n, y_n, z_n, v_n

Implemented by MinkovskiEngine from NVIDIA



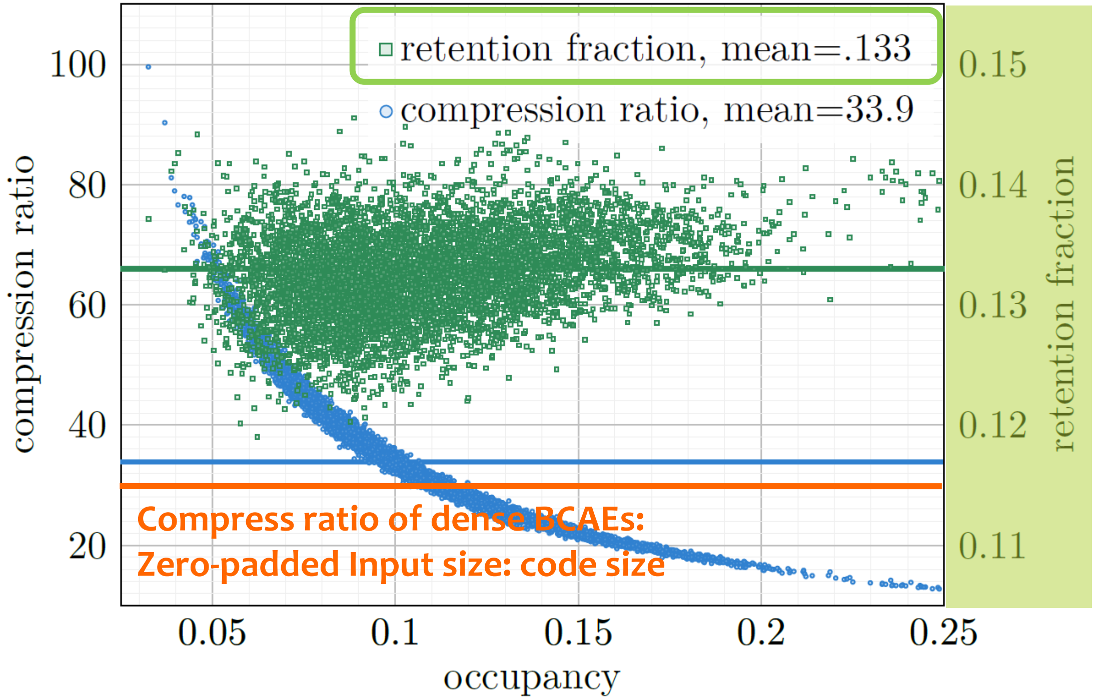
Reconstruction Accuracy



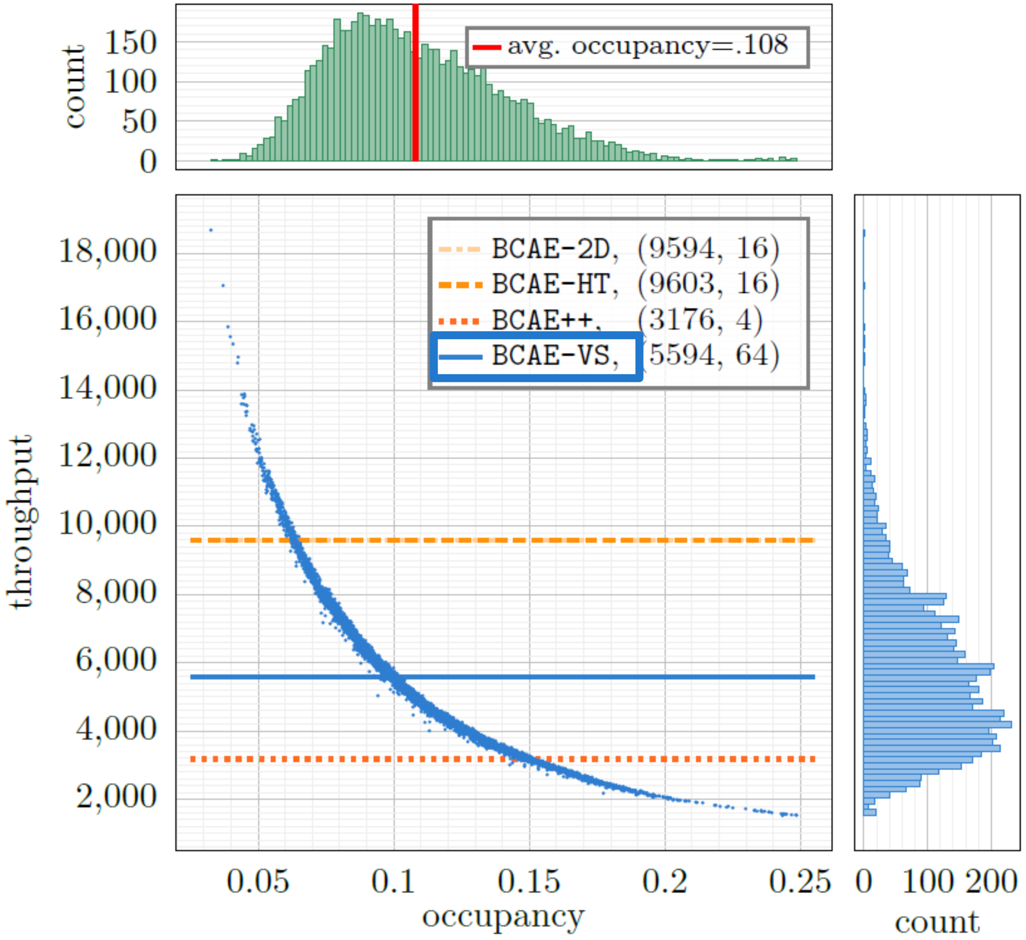


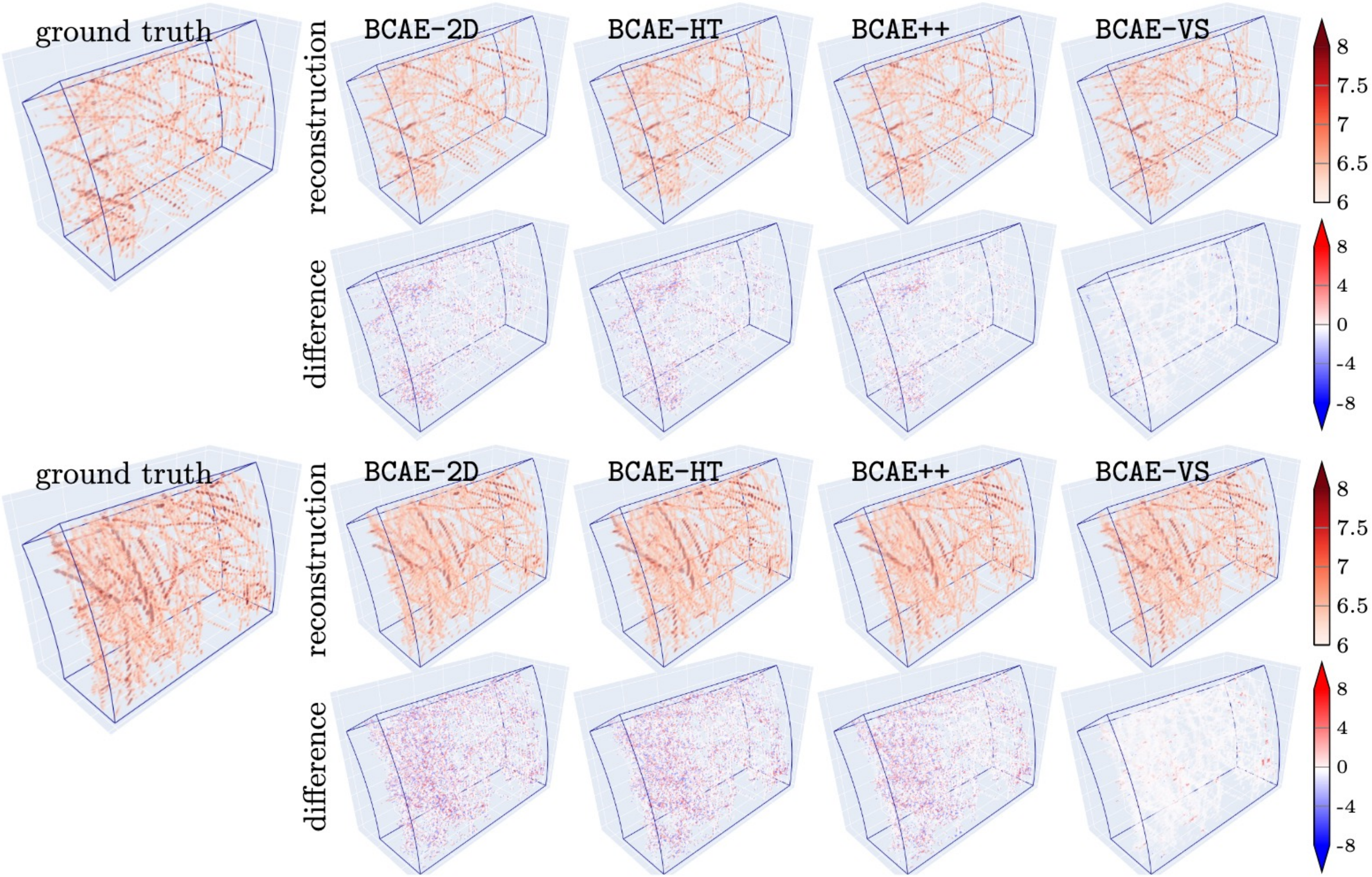
Variable Compression Ratio and Throughput as Function of Occupancy

fraction of signals saved to the persistent storage (~1/compression_ratio for ZS data)



throughput = number of TPC wedges processed by one GPU per second. GPU we used is NVIDIA 6000 ADA.



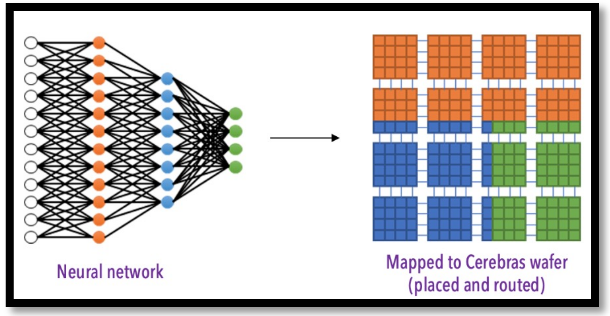

ground truth
reconstruction
difference
BCAE-2D
BCAE-HT
BCAE++
BCAE-VS
8
7.5
7
6.5
6
8
4
0
-4
-8
ground truth
reconstruction
difference
BCAE-2D
BCAE-HT
BCAE++
BCAE-VS
8
7.5
7
6.5
6
8
4
0
-4
-8

Performance comparison

model	comp. ratio ↑	reconstruction performance L_1 ↓	L_2 ↓	PSNR ↑	recall ↑	precision ↑	efficiency encoder size	throughput ↑
BCAE-2D	31	.152	.862	20.6	.907	.906	169k	**9.6k**
BCAE-HT (3D)	31	.138	.781	20.8	.916	.915	9.8k	**9.6k**
BCAE++ (3D)	31	.112	.617	21.4	.936	.934	226k	3.2k
BCAE-VS	**34**	**.028**	**.089**	**26.0**	**.988**	**.996**	**382**	5.6k

Huang, Y., Go, Y., Huang, J., Li, S., Luo, X., Marshall, T., ... & Yoon, B. J. (2024). Variable Rate Neural Compression for Sparse Detector Data. *arXiv preprint* arXiv:2411.11942.

Summary



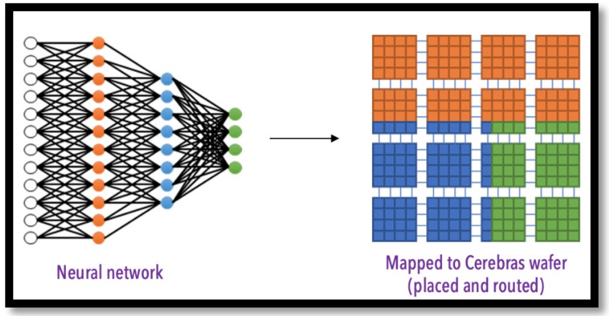
- Bicephalous Convolutional Auto-encoder (BCAE) for sparse TPC data
- Faster BCAE-2D with Encoder-Decoder tradeoff
- Variable compression rate and computation with BCAE-VS.
- (Future) Improve BCAE-VS in low occupancy region
- (Future) Improve throughput of BCAE-VS on GPU and other hardware
- (Future) noise rejection, tracking efficiency, etc.

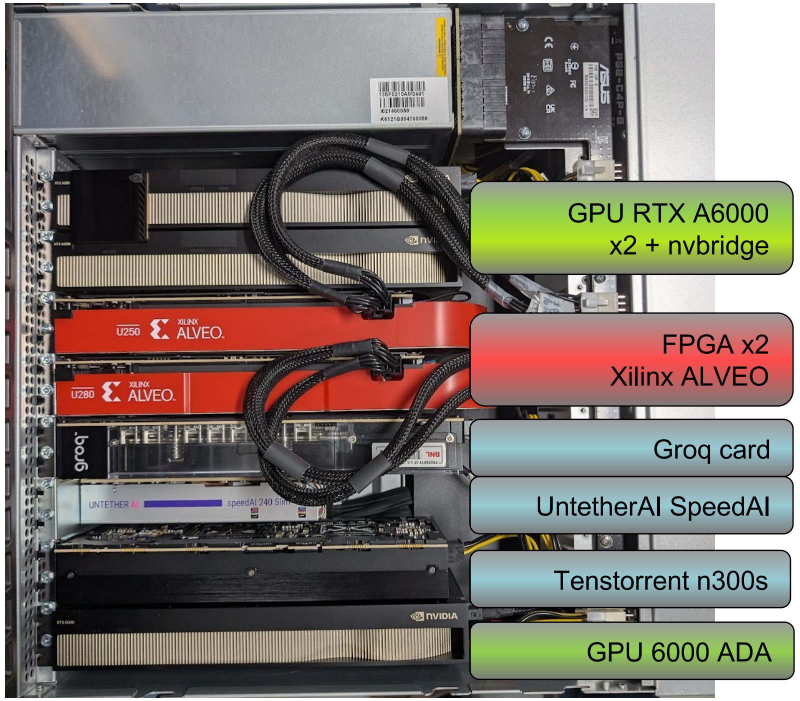
Source Code:

- BCAE https://github.com/BNL-DAQ-LDRD/NeuralCompression
- BCAE-2D https://github.com/BNL-DAQ-LDRD/NeuralCompression_v2
- BCAE-VS https://github.com/BNL-DAQ-LDRD/NeuralCompression_v3

Summary

- Bicephalous Convolutional Auto-encoder (BCAE) for sparse TPC data
- Faster BCAE-2D with Encoder-Decoder tradeoff
- Variable compression rate and computation with BCAE-VS.
- (Future) Improve BCAE-VS in low occupancy region
- (Future) Improve throughput of BCAE-VS on GPU and other hardware
- (Future) noise rejection, tracking efficiency, etc.





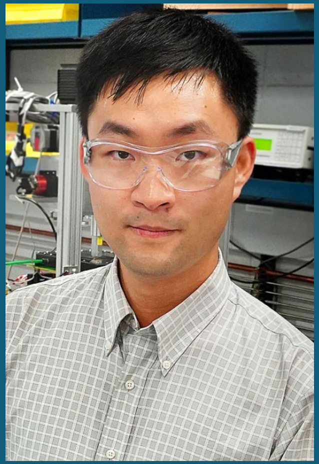
Feel free to try on your streaming detector data! Source Code:

- BCAE https://github.com/BNL-DAQ-LDRD/NeuralCompression
- BCAE-2D https://github.com/BNL-DAQ-LDRD/NeuralCompression_v2
- BCAE-VS https://github.com/BNL-DAQ-LDRD/NeuralCompression_v3

Thank you!

ありがとうございます

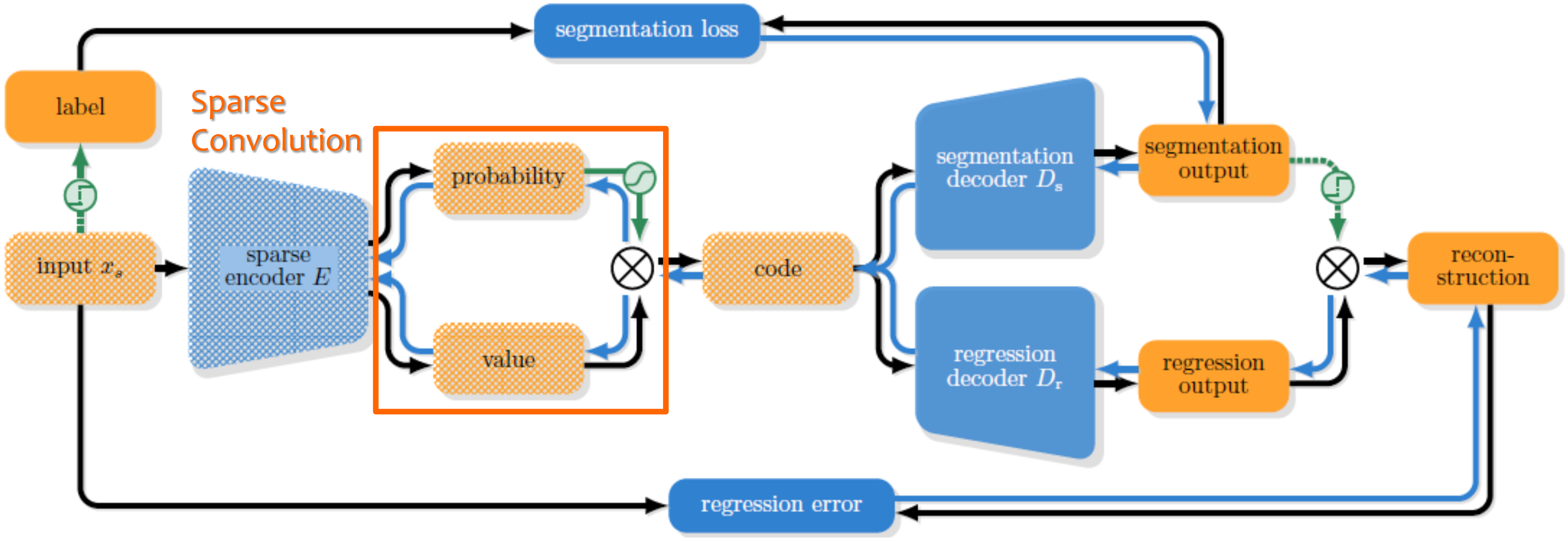
Yihui “Ray” Ren (yren@bnl.gov)



Yi Huang, Yihui “Ray” Ren, Yeonju Go, Xihaier Luo, Shuhang Li, Thomas Marshall, Joseph D. Osborn, Christopher Pinkenburg, Evgeny Shulga, Shinjae Yoo, Byung-Jun Yoon, **Jin Huang** (PI)

BCAE-VS

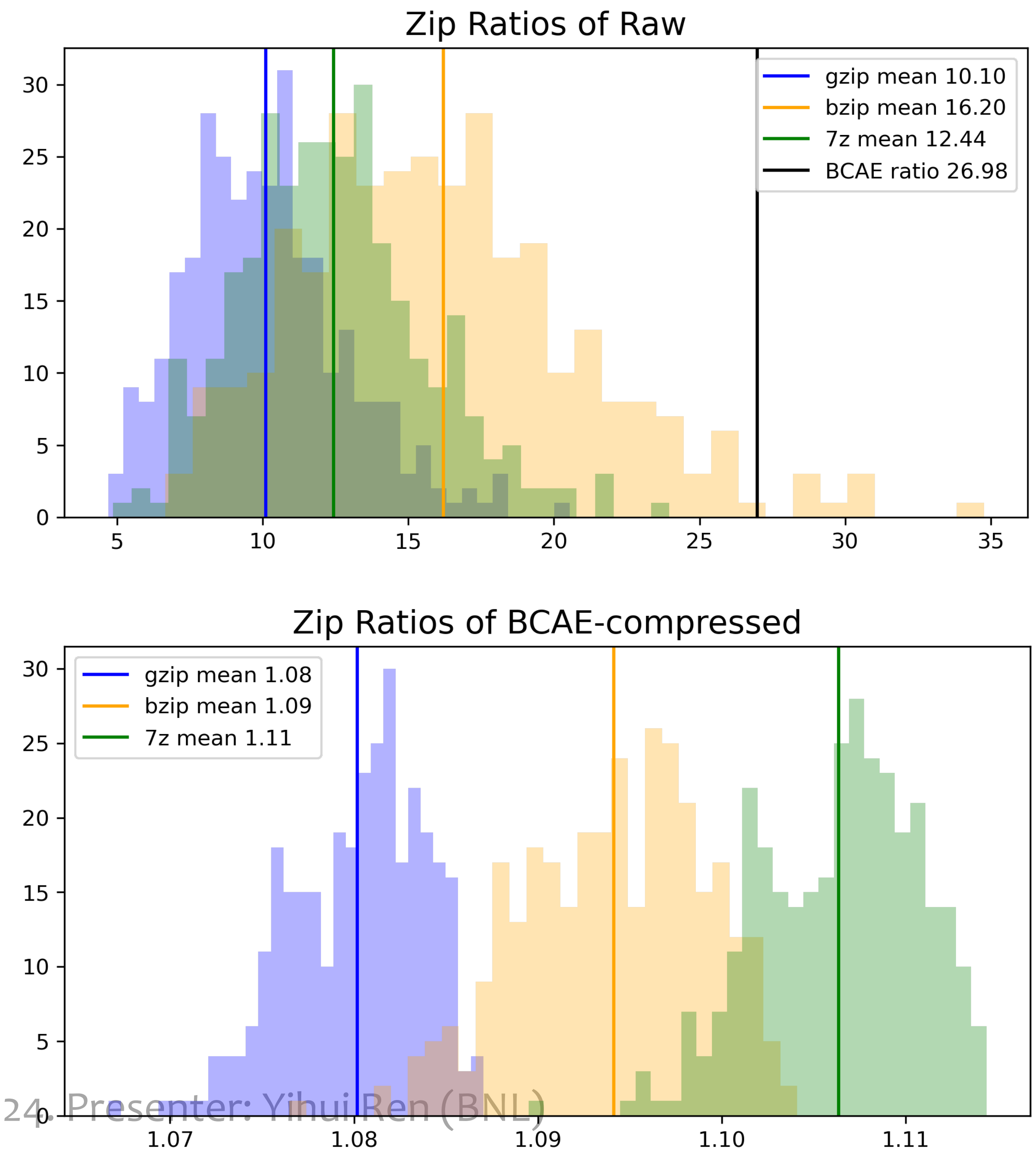
Variable ratio Compression for Sparse input



Sanity Check with lossless compressions

- Compare to lossless compression tools. More than twice compression ratio.
- Compressed code of BCAE can not be compressed further.

How is our model comparing to other lossy compression algorithms?



Real-time AI accelerator

gpuserver0.sphenix.bnl.gov

Gpuserver0 upgrade yesterday

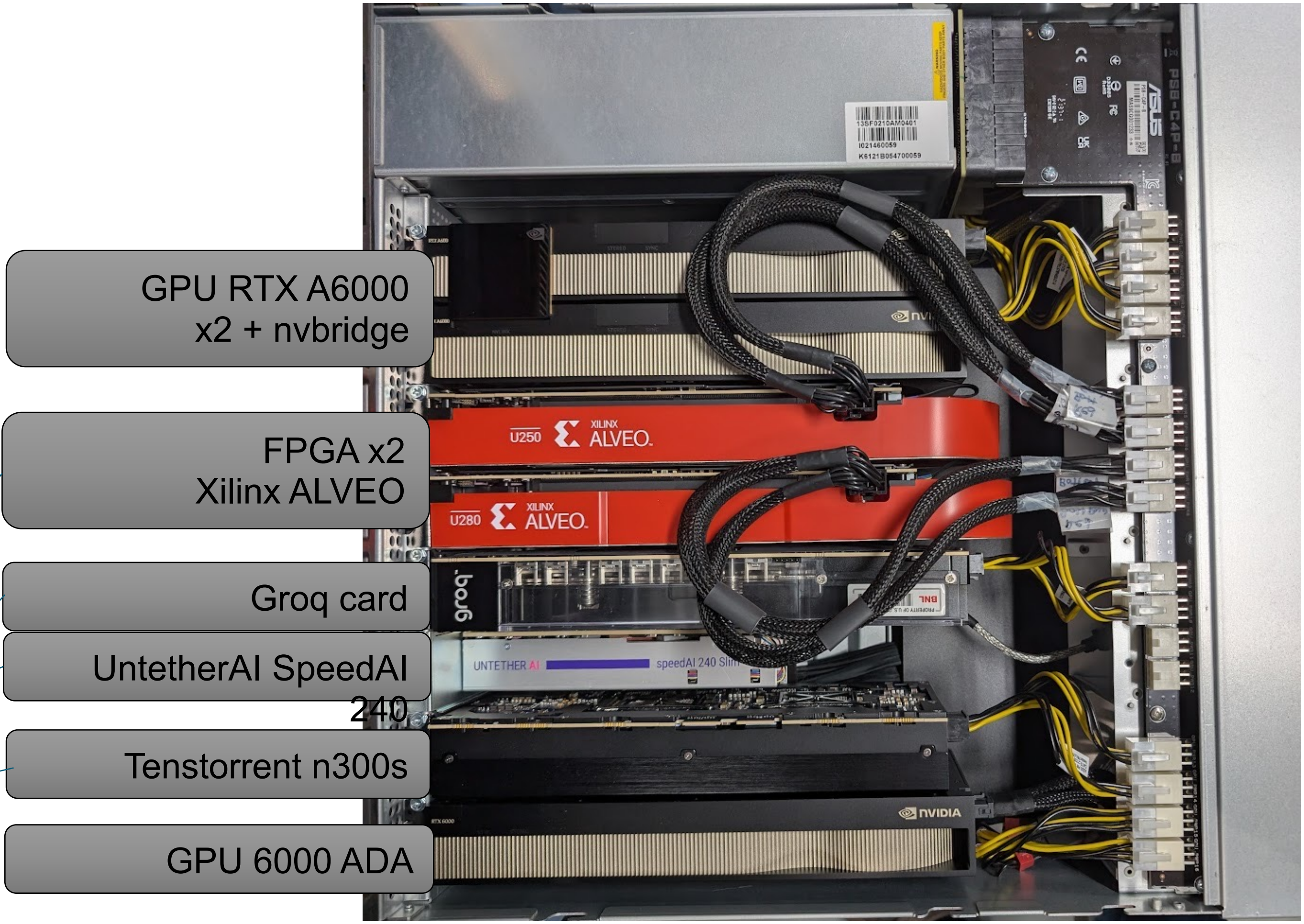
L40S removed for RMA

Three AI chips:

- Groq: TPU based
- UntetherAI: INT8 optimizes
- Tenstorrent: RISC-V cores

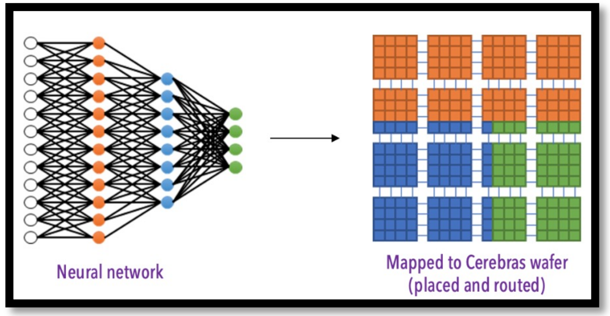
More photos:
https://photos.app.goo.gl/z3AnNhfrd4bqZTeN6

```
jinhuang@gpuserver0:~$ sudo lspci | grep cc
41:00.0 Processing accelerators: Xilinx Corporation Device d004
61:00.0 Processing accelerators: Xilinx Corporation Device d00c
81:00.0 Processing accelerators: Device 1e67:0004 (rev 01)
a1:00.0 Processing accelerators: Groq TSP100 Tensor Streaming Processor
c1:00.0 Processing accelerators: Device 1e52:401e (rev ff)
```



Summary

- Bicephalous Convolutional Auto-encoder (BCAE) for sparse TPC data
- Faster BCAE-2D with Encoder-Decoder tradeoff
- Variable compression rate and computation with BCAE-VS.
- (Future) Improve BCAE-VS in low occupancy region
- (Future) Improve throughput of BCAE-VS on GPU and other hardware
- (Future) noise rejection, tracking efficiency, etc.



Source Code:
- BCAE https://github.com/BNL-DAQ-LDRD/NeuralCompression
- BCAE-2D https://github.com/BNL-DAQ-LDRD/NeuralCompression_v2
- BCAE-VS https://github.com/BNL-DAQ-LDRD/NeuralCompression_v3