Compression of Scientific Data with SZ

Franck Cappello

Argonne National Laboratory

University of Illinois at Urbana Champaign

With:

Sheng Di, Robert Underwood, Julie Bessac, Dingwen Tao, Xin Liang, Kai Zaho, Xiaodong Lu, Jon Calhoun, Hanqi Guo, Jiannan Tian, Jinyang Liu, Codi Rivera, Ali Murat Gok and more…

Argonne NATIONAL LABORATORY

Lossy compression of scientific data

- Consist in reducing scientific data volume by leveraging correlations and reducing precision (lossless compression does not reduce scientific data enough)

- Compression ratios (with current compressors) vary depending on use-cases, typically:
  - CR=5 for hard to compress dataset and demanding data/analysis quality preservation
  - CR=10-100 for scientific data presenting high correlation and medium data/analysis quality preservation
  - CR=x100 for visualization (low data/analysis quality preservation)

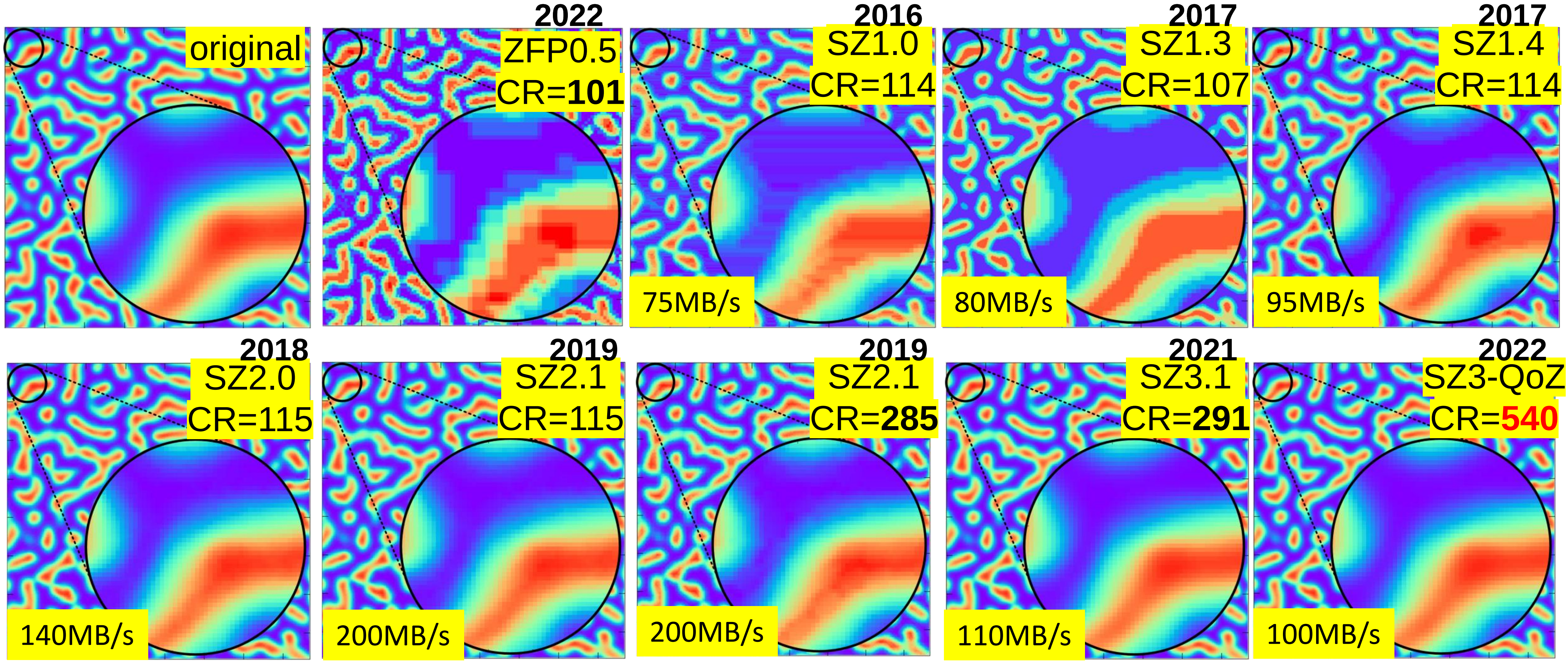
- Goal: **keep the same science** (satisfy user's quality requirements WRT QoIs – features)
  - WARNING: You will see images because this is the easiest way to show distortion but compression of scientific data is NOT only for images

- Getting significant traction in the scientific community (climate, cosmology, seismic, etc.), IoT community as well (sensors, EKG)

Argonne NATIONAL LABORATORY

Huge Progress in performance in the past 5-6 years

Evolution of SZ compression quality and performance using a large-eddy simulation of multicomponent flows with turbulent mixing: Miranda - density field.



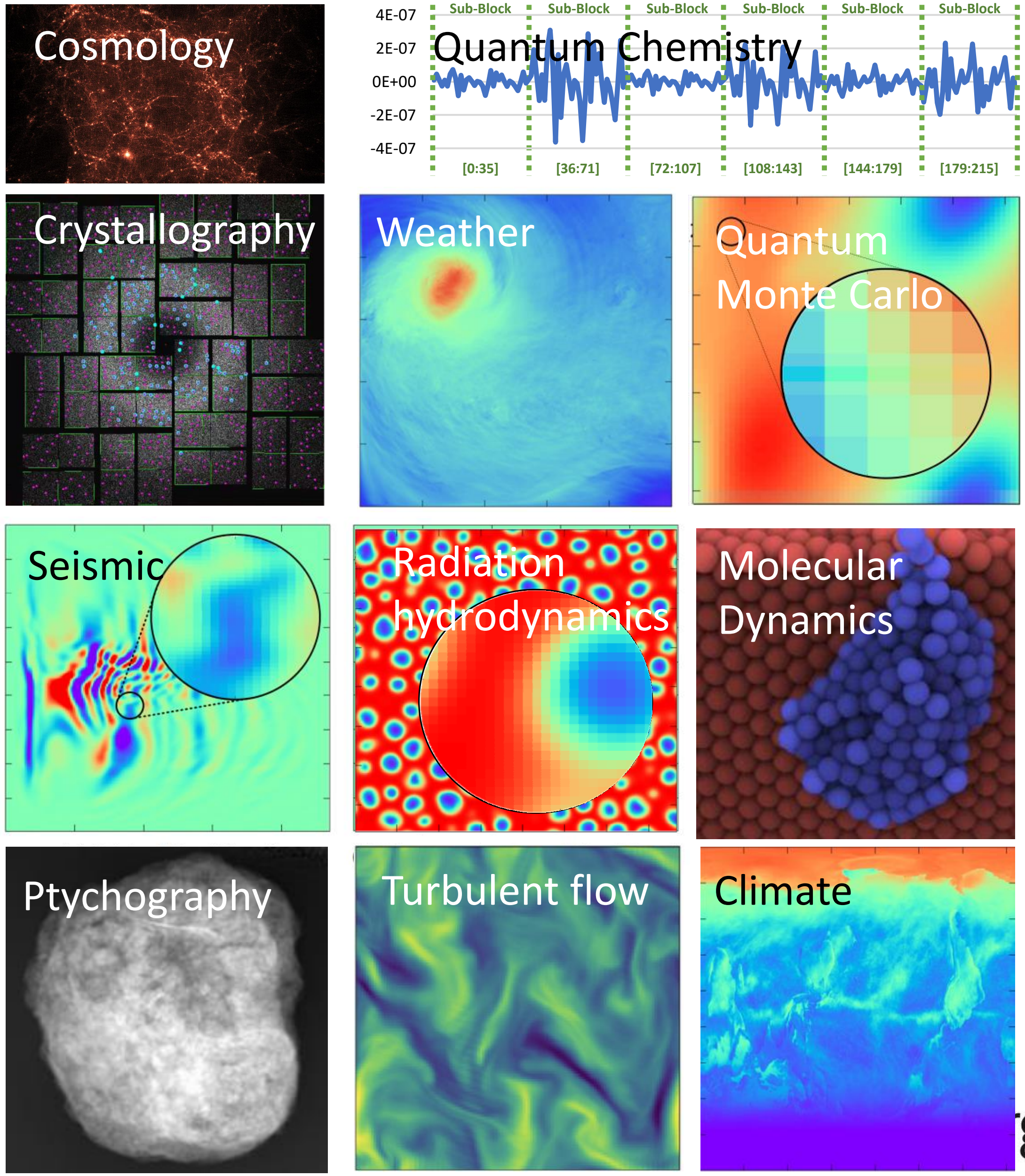
Visualization of Miranda - density data for SZ's different versions (EB: VRAE 1E-2), Performance on single core CPU (Intel Broadwell)

SZx compresses at 300GB/s on NVIDIA A100 → Bottleneck is not compression but PCIe

Many Applications Domains

We worked directly for all these domains to develop SZ

- Climate
- Combustion
- Cosmology
- Deep Learning
  - Activation data
  - Model coefficients
  - Training data
- Extreme Weather
- Fusion Energy
- Hydrodynamics
- IoT
- **Light Sources (LCLS, APS, etc.)**
- Materials Science
- Molecular Dynamics
- Quantum Chemistry
- Quantum Circuit Simulation
- Seismic Imaging



EXASCALE COMPUTING PROJECT

Argonne NATIONAL LABORATORY

Many Use-Cases

We are seeing an increasing diversity/number of use-cases

"Classic" use-cases:

1) Visualization
2) **Reducing storage footprint** (offline compression)
3) Reducing I/O, communication time (on-line, in-situ compression)

Recently identified use-cases:

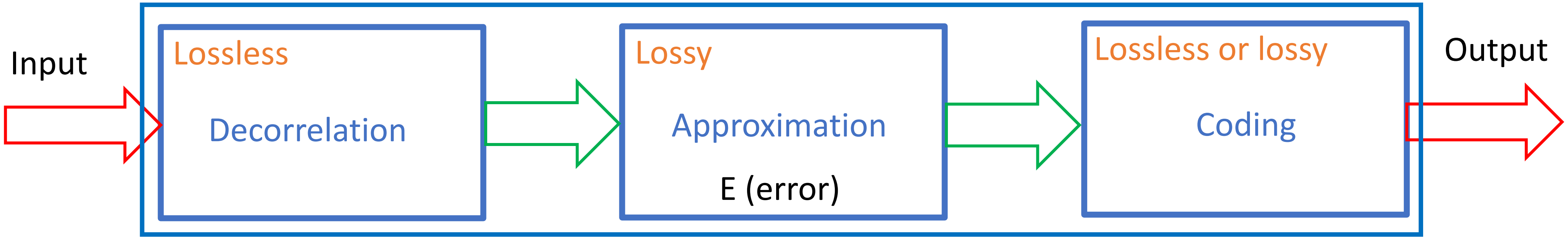
4) **Reducing streaming intensity** (recent for generic floating-point compressors)
5) Lossy checkpoint/restart from lossy state
   - reduce checkpoints footprint on storage – adjoint, accelerate checkpointing
6) Re-computation Avoiding by reducing the memory footprint → GAMESS
7) Running larger simulations by reducing the memory footprint
8) Accelerating CPU/GPU – memory transfer
9) Reduce DNN model size
10) Accelerate training (I/O read time) of DNNs

SZ has been evaluated for all these use-cases

Cappello, F., Di, S., Li, S., Liang, X., Gok, A. M., Tao, et Al., Use cases of lossy compression for floating-point data in scientific data sets. *The International Journal of High Performance Computing Applications*, *33*(6), 1201–1220, 2019

General Principle of Error Bounded Lossy Compression

Typical design of a lossy compressor for scientific data



Most of the researches in the past 5 years (Transforms, Predictors, SVD, etc.)

This is where compression error is controled:

- **Point wise error bounds**
- **Statistical metrics**
- **Feature preservation**

Very well known 70 years of Shannon Theory (still some research on high performance codding)

QoIs: Quantities of interest

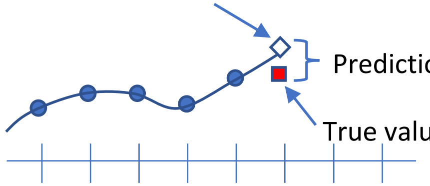
SZ as a Software (Responds to ECP users)

https://szcompressor.org

- **Compress/decompress by blocks** for nearly **random-access decompression**
- Can compress **1D, 2D, 3D datasets.** and unstructured datasets as 1D
- Multiple // implementations: CPU Core (Vector Instructions), Multi-core (OpenMP), GPU (Cuda, Kokkos, *HIP*, DPC++*),* FPGA (proto)
- Integration in **HDF5, ADIOS** and **PnetCDF**
- **Production quality**. test infrastructure, scripts for regression testing

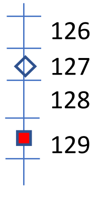
Input → Pre-Processing (log trans.) → Decorrelation: Prediction → Approximation: Quantization → Coding: Entropy coding → Post Processing: Z-standard, Gzip → Output

Floating point stream →



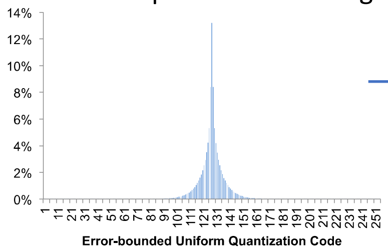
Floating point →

Quantization of the prediction error



Integer (bin #) →

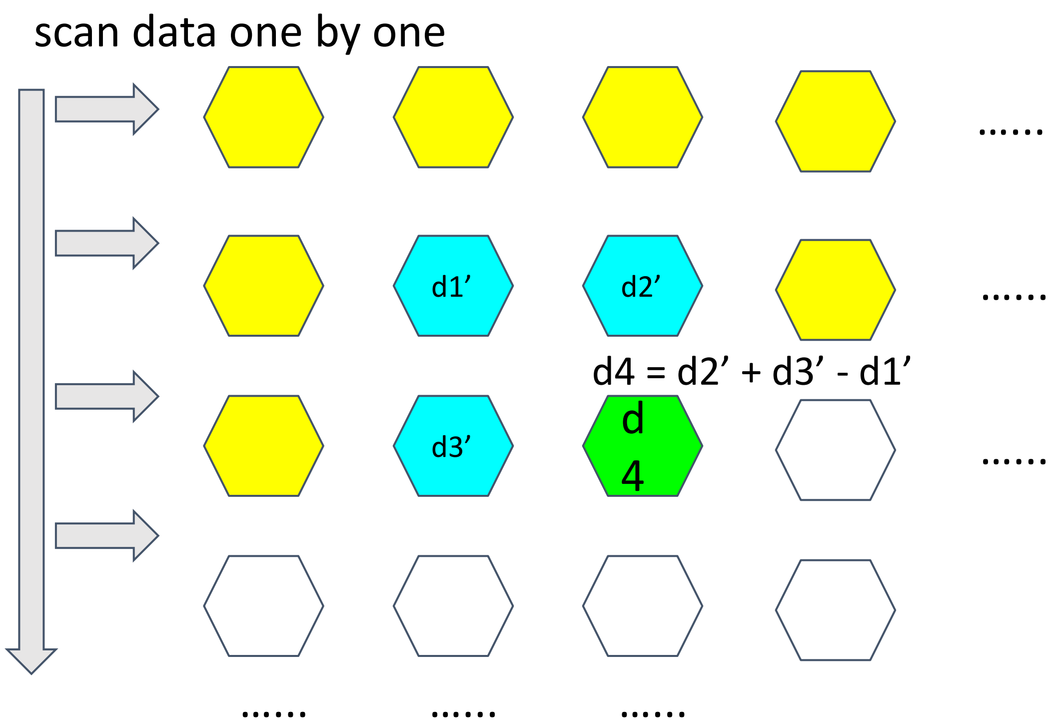
Reduce the number of bits to represent each integer



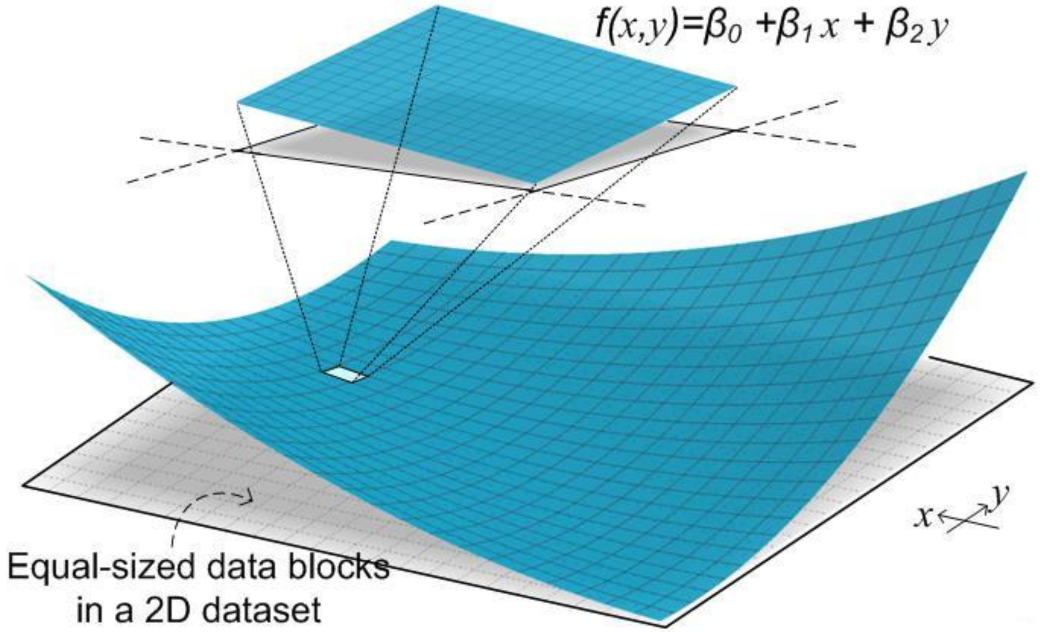
Bit stream →

Examples of Predictors

Lorenzo



Linear-regression



the constructed hyperplane must be based on “decompressed” coefficients

And many others: Multi-level interpolation, pattern based, DNN, Wavelet, etc.
For 1D, 2D, 3D and 4D (3D + time) datasets.

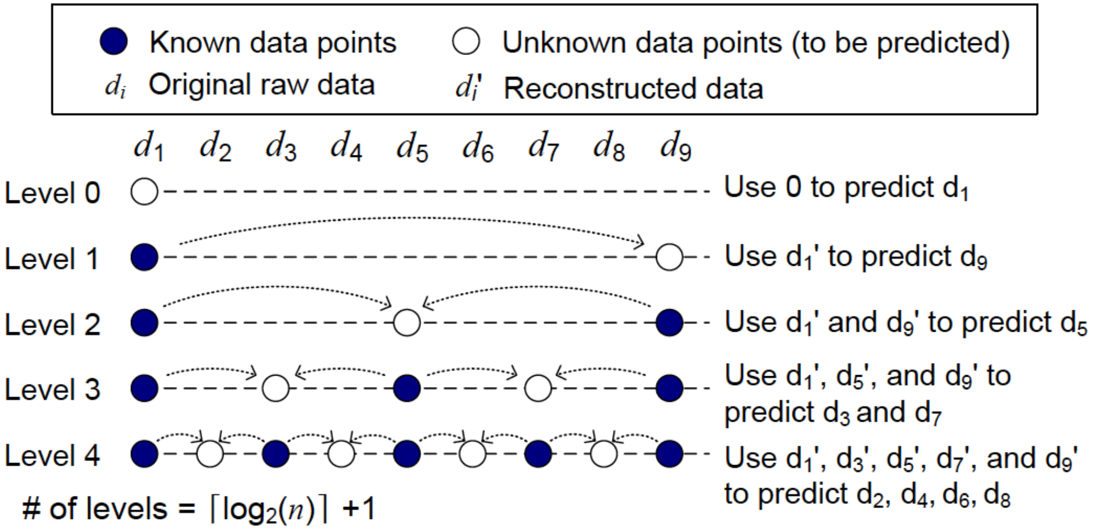
Argonne NATIONAL LABORATORY

Example: SZ Interpolation based Predictor

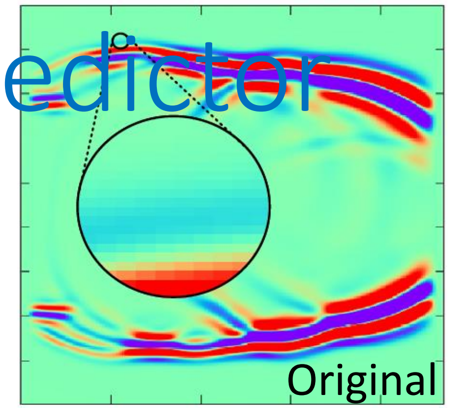
Predictor based on multilevel, multidimensional tri-cubic spline interpolation

Spline method	*Prediction Value* p_i
Linear spline	$p_i = \frac{1}{2}d_{i-1} + \frac{1}{2}d_{i+1}$
Cubic spline	$p_i = -\frac{1}{16}d_{i-3} + \frac{9}{16}d_{i-1} + \frac{9}{16}d_{i+1} - \frac{1}{16}d_{i+3}$

1D case (linear spline):

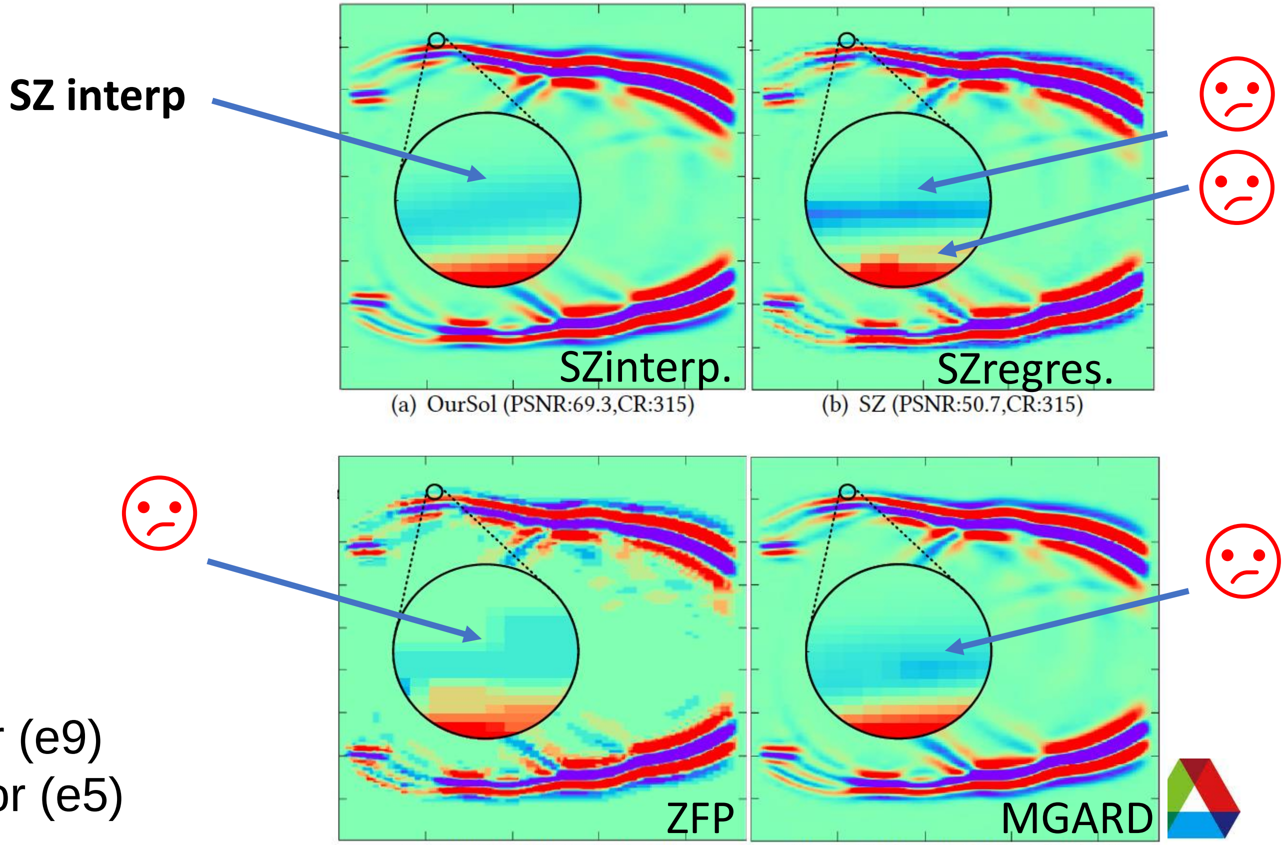


At level 0, 0 to predict $d1$, → Store quantized error (e0)
At level 1, $d1$+e0 to predict $d9$, → Store quantized error (e9)
At level 2, $d1$+e0 and $d9$+e9 to predict $d5$, → Store error (e5)
…



CR:~315

Figure 25: Visualization of RTM, original data



(a) OurSol (PSNR:69.3,CR:315)
(b) SZ (PSNR:50.7,CR:315)
(c) ZFP (PSNR:51.7,CR:258)
(d) MGARDx (PSNR:62.5,CR:310)

Generic with App Specific Performance: Customization

Goal: reach performance (ratio, speed, accuracy) as close as possible to application specific data reduction without requiring expensive development/maintenance/update costs.

Too specific:
Expensive to Develop, Maintain, Update

Multi-algorithms
(per block automatic selection of predictors)
SZ 2.0

Hybrid/Autotuning
(per block automatic selection of decorrelation)

Customizable
(composition of compression pipeline)
SZ 3.0

Fully Generic
Few parameters:
Error bound
-Bit grooming
-Digit Shuffling

Fully Specific

Parametrizable
(1 algorithm)
ZFP

Meta-Compressor
(per block automatic selection)
SZ-ZFP Hybrid

Preconditioners
(Add specific stages to separate data, improve/complement generic compressors)

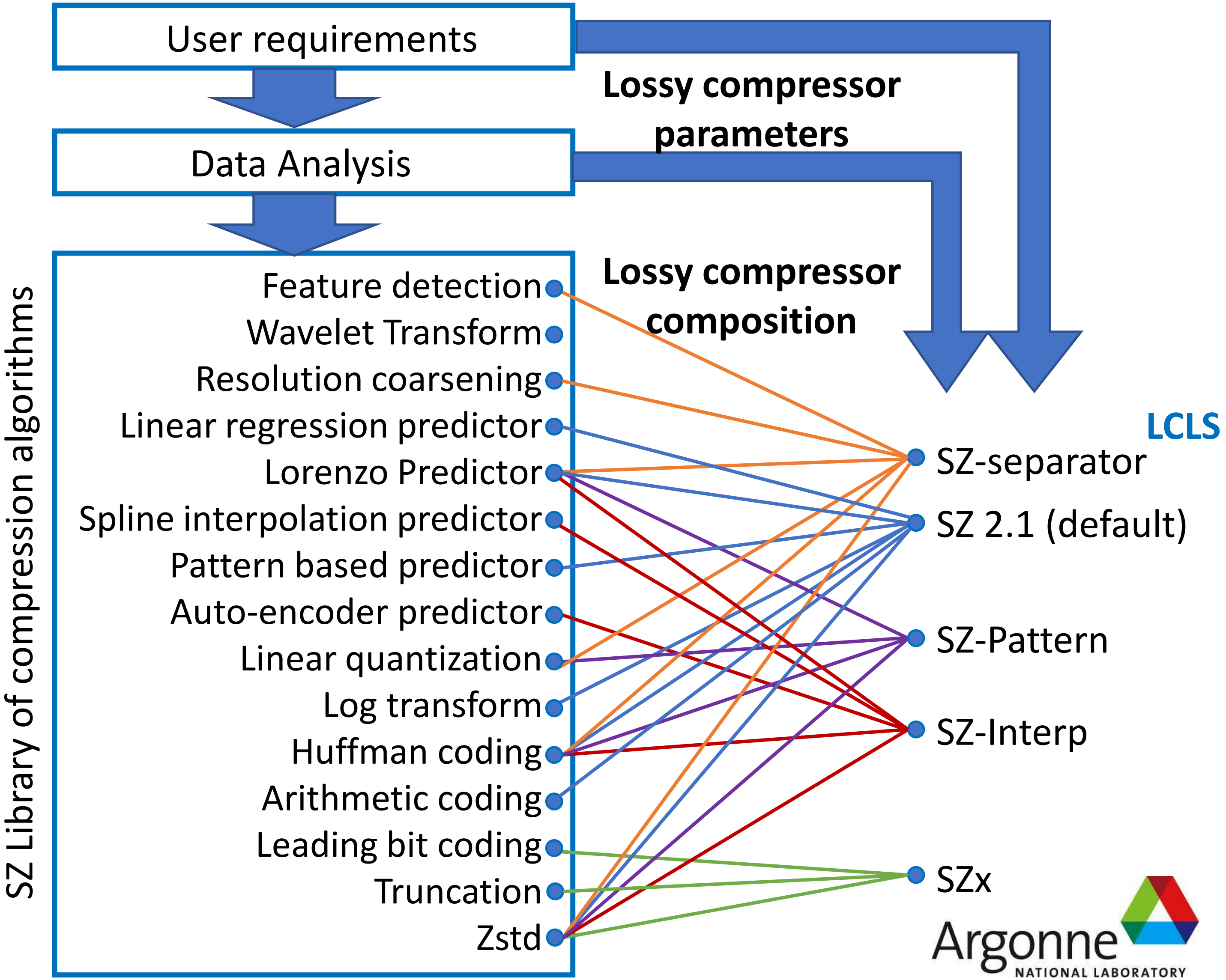
Higher performance
(ratio, speed, accuracy)

Argonne NATIONAL LABORATORY

What makes SZ3 different: a Highly **Modular**/ **Customizable** Compression Framework

SZ 3 (C++) library of algorithms for lossy compression and examples of SZ compressors built from the library of algorithms.

To compose and tune a compression pipeline we analyze the data to compress and user requirements in compression speed, ratio and accuracy.



Example: Cosmology 1/2 (Storage Footprint Reduction)

HACC: N-body problem with domain decomposition, medium/long-range force solver (particle-mesh method), short-range force solver (particle-particle/ particle-mesh algorithm).

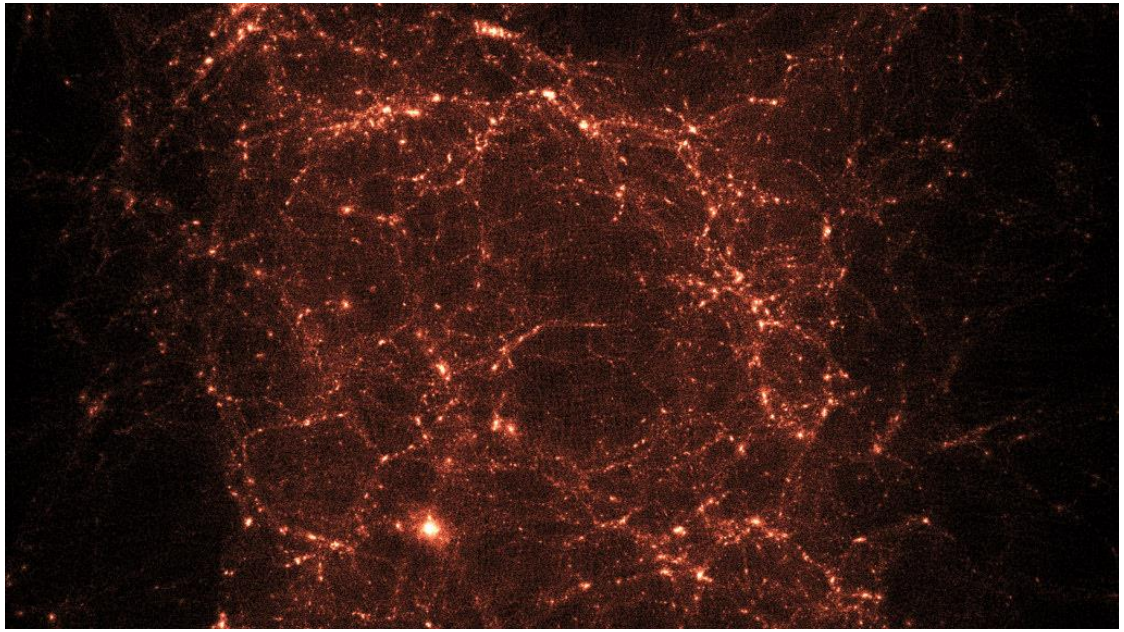
Particle dataset: 6 x 1D array (x, y, z, vx, vy ,vz)

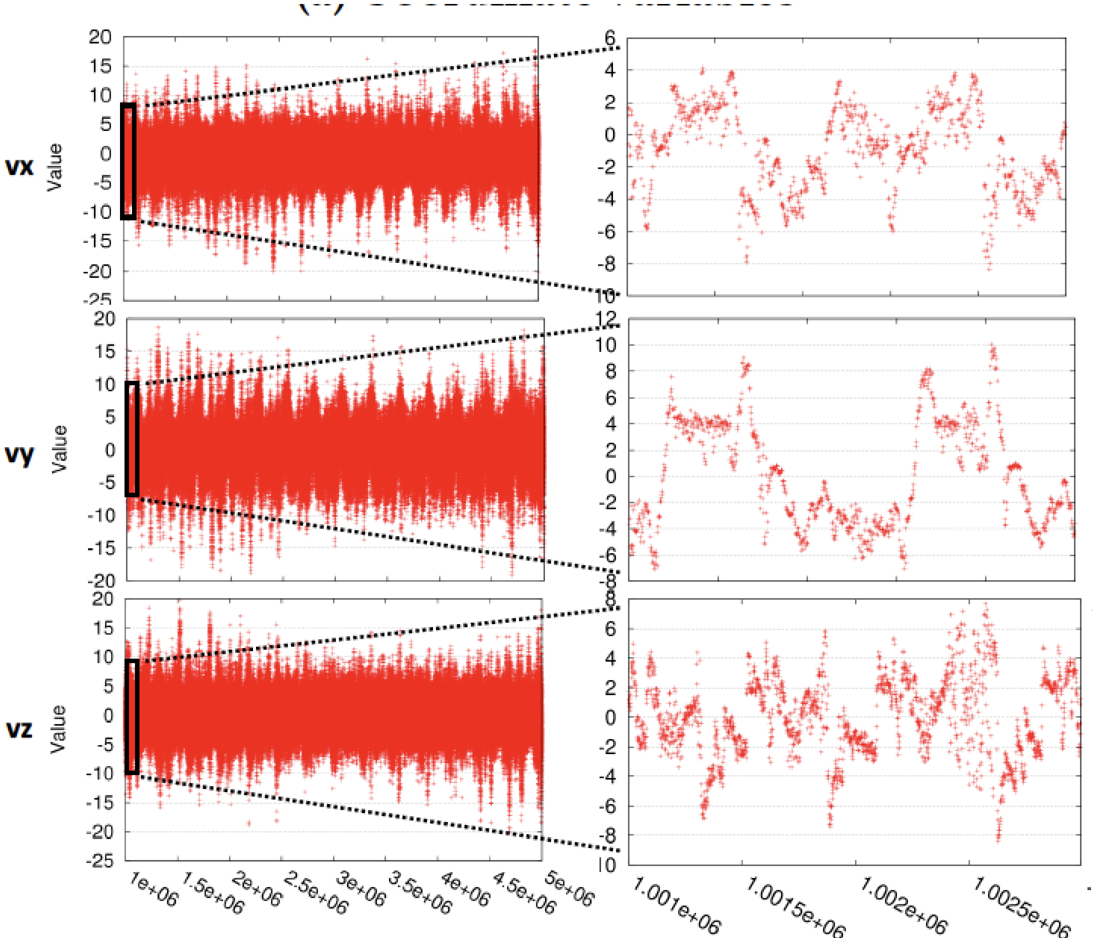
SZ 2.0: **CR ~5** (~6bits/value) at 10^{-3} error bound

Preferred error controls:

- Point wise max error (Relative) bound
- Absolute (position), Relative (Velocity)

ANL: Cosmological Simulations for Large-Scale Sky Surveys





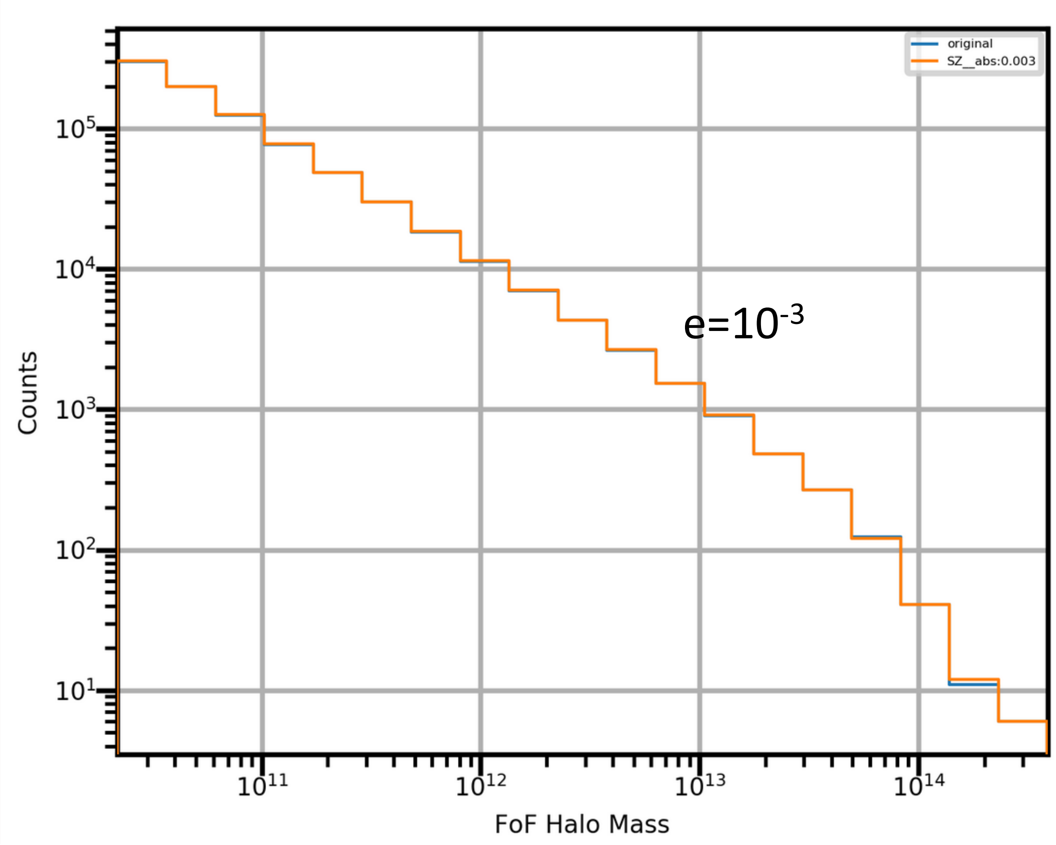
Example: Cosmology 2/2

HACC

Results validation

3kpc absolute error bound
(particle position)

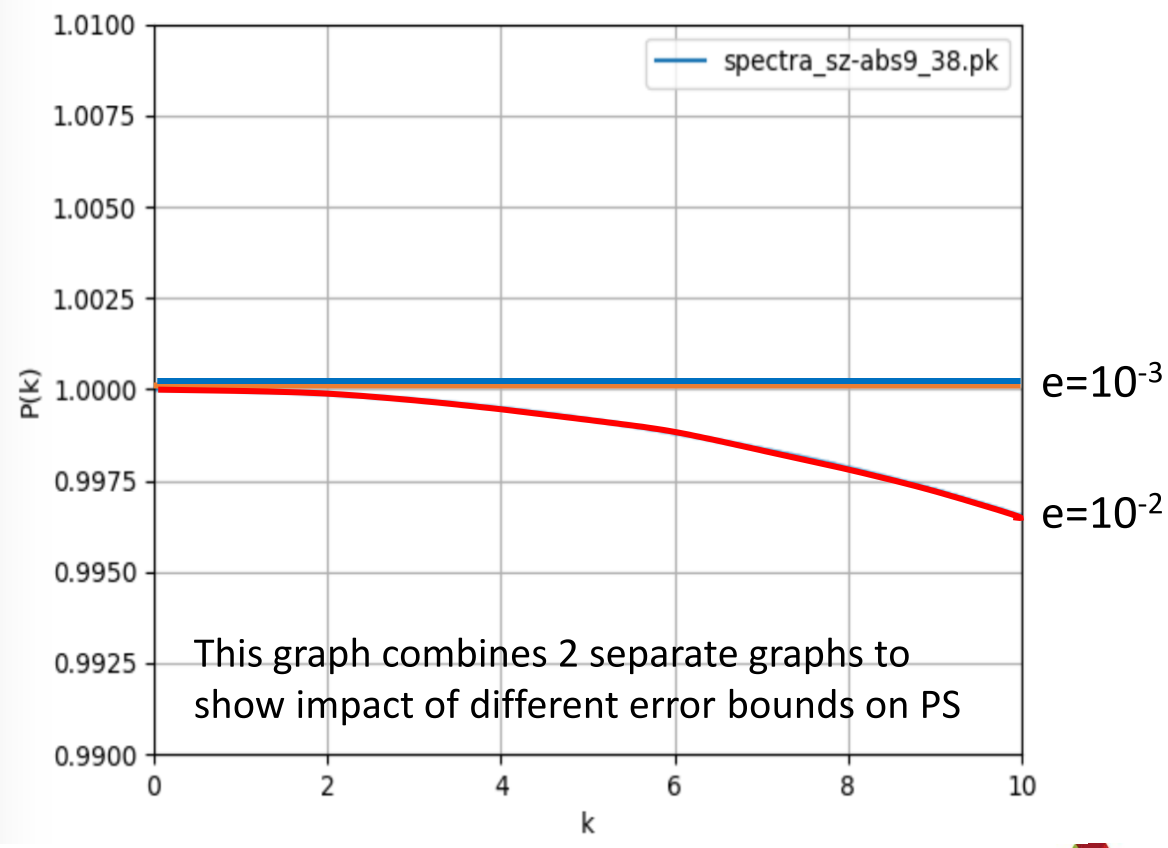
Original:
SZ (e=10^{-3}):
SZ (e=10^{-2}):



Friends of Friends halo mass distribution

Results validation

3kpc absolute error bound
(particle position)



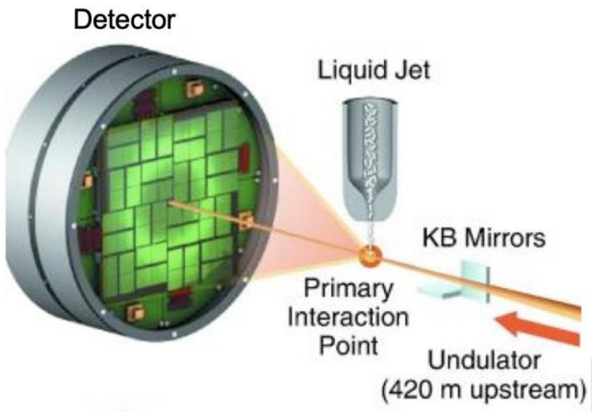
Power Spectrum

S. Li, S. Di, X. Liang, Z. Chen, F. Cappello, Optimizing Lossy Compression with Adjacent Snapshots for N-body Simulation Data, **IEEE BigData 2018**

Example: Crystallography (Streaming intensity)

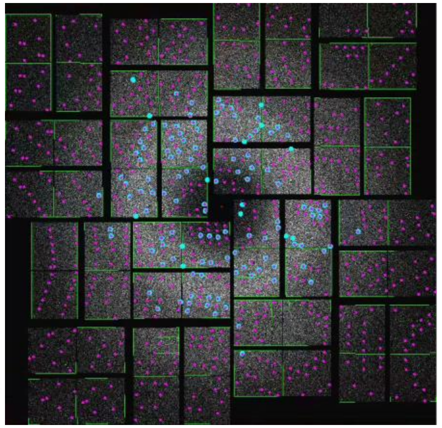
Chuck Yoon: (Stanford, LCLS)

1: X-ray Beam



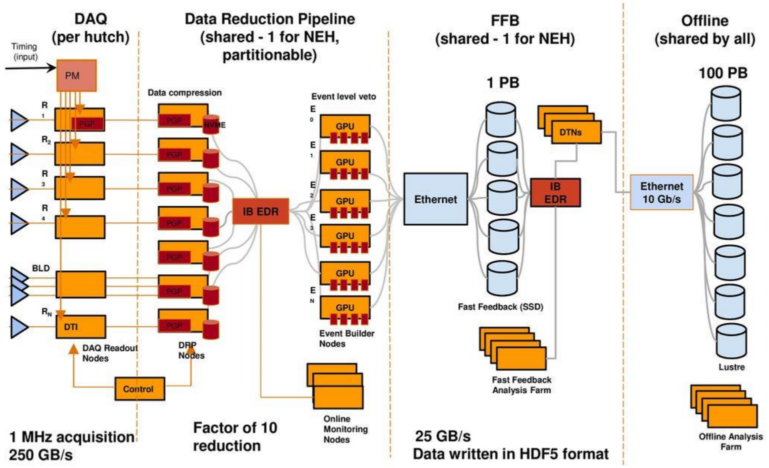
Diffraction before destruction
Number of pulses/sec: 120
Millions of diffraction patterns from crystals

2: Diffraction



LCLS-II Data System

3: Reduction



Context: LCLS II. Goal: Definition of reduction method
Detector produces:

- 2D images @ 250GB/s
- 4M pixel/event unsigned integers, in binary XTC2 format

Compression objectives: **CR of 10 or more** with error bound @ 500 MB/s/core
→ **RoiBinSZ** algorithm (regions of interest extraction + background binning + SZ background compression)

Data from detector

RoiBinSZ compression pipeline

- Peak finder
- Mask regions of interest
- Extract background
- Low pass filter (binning e.g. 2x2)
- Compress background (SZ 2.1)
- Rebuild image

- Indexing
- Merge
- Phase
- Refine

Image

Crystallography: First Level of Analysis Distortion: Indexing

EXASCALE COMPUTING PROJECT

Chuck Yoon: Stanford

Roibin SZ on Se-SAD SFX Dataset (Selenium)

selenobiotinyl-streptavidin on a cspad detector

- Number of hits: An image with at least 15 peaks is considered a hit
- Number indexed: Number of crystals extracted from hits
- Rsplit: measure precision of averaged intensities/amplitudes
- CCano: The correlation coefficient of the Bijvoet differences of acentric reflections
- CC1/2: Pearson correlation coefficient.
- Rwork: measure of the agreement between the crystallographic model and the experimental X-ray diffraction data
- Rfree: Rwork computed on a small, random sample of data
- Map-model CC: cross-correlation between electron density map and model.

	Original	**Riobin SZ**
Total compression ratio	1	**70.65**
Number of hits	744,150	744,150
Number indexed	255,065	255,918
Rsplit ↓	7.58%	**7.08%**
CC1/2 ↑	0.997	**0.997**
CCano ↑	0.087	**0.104**
Rwork ↓	0.206	**0.199**
Rfree ↓	0.231	**0.223**
Map-model CC ↑	**0.81**	0.8

↑: higher the better ↓: lower the better

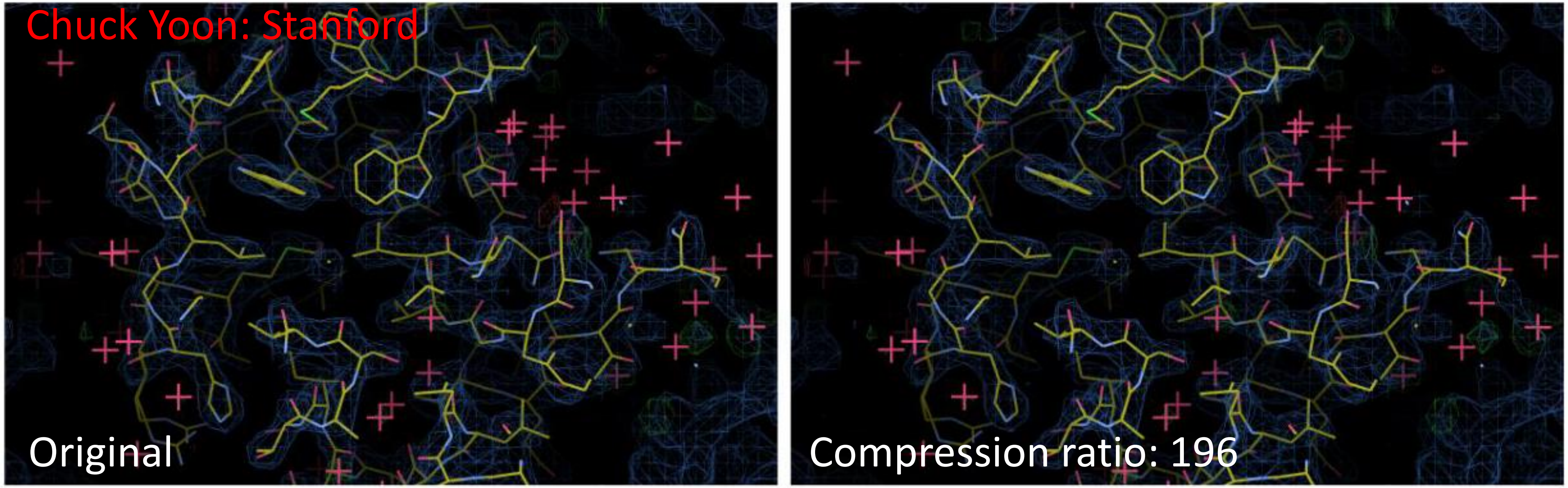
Argonne NATIONAL LABORATORY

Crystallography: Final Level of Analysis

Distortion: Protein Reconstruction

Reconstruction of Electron Densities Lysozyme

Very important role in our immune system: breaks up (digests) components of the cell walls of bacteria.



Lysozyme on a jungfrau4m detector

(a) original (b) roibin-sz

The data on the right is 196x smaller (or 631× if also using Non-Hit Rejection)

gonne
NATIONAL LABORATORY

Example: Ptychography (Storage Footprint Reduction)

Tekin Bicer (DSL and XSD)

Beamline Scientists: Junjing Deng, Jeff Klug and others
Compression and reconstructions: Sheng Di, Tekin Bicer

Timing: Bebop cluster, Intel Xeon E5-2695v4 (1 core).

Original dataset: Catalyst Particle
Compressed with SZ2.1 (not Riobin SZ)
Single scan (diffraction patterns): 1856x1030x514
Compressed 1856 images of 514x1030 pixels.

For the spatial compression, the dataset is treated as a 3D dataset, so the predictor adopts a 3D Lorenzo + 3D Linear regression;

For the temporal compression, the compressor predicts each data point only based on its temporal dimension

Tested absolute error bound from 2 to 64.
Absolute error bound of 2 translates to (+/-) 2 photon count error on the detector.

PSNR computed from the diffraction patterns (not reconstruction result)

Absolute error bounds

RATIO	2	4	8	16	32	64
Spatial	72.9	**97.2**	117.7	144.7	147.2	181.1
Temporal	90.2	123.2	**245.1**	307.3	354.4	465.1

Timing (secs, comp/decomp)	2	4	8	16	32	64
Spatial	18.6/ 8.3	18.5/ 7.6	18.6/ 7.4	18.8/ 7.1	18.5/ 7.4	17.5/ 7.6
Temporal	28.1/ 16	29.4/ 15.6	27.8/ 15	27.7/ 14.9	27.6/ 14.9	29/ 14.7

GB/s (comp)	2	4	8	16	32	64
Spatial	201.5	**202.6**	201.5	199.3	202.6	214.1
Temporal	133.3	127.4	**134.8**	135.3	135.8	129.2

PSNR	2	4	8	16	32	64
Spatial	200.1	**196.7**	192.7	188.5	180.5	175.7
Temporal	194.2	187.9	**185.0**	181.9	167.9	165.6

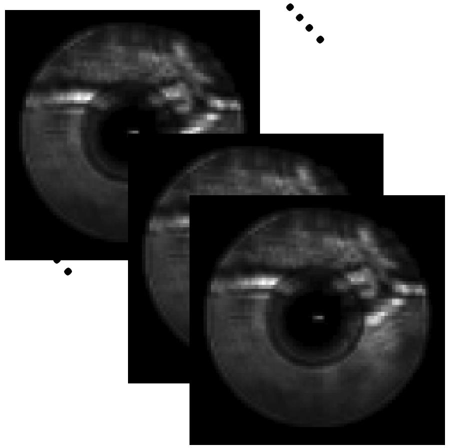
Ptychography: Reconstruction from Diffraction Pattern

Tekin Bicer (DSL and XSD)

Beamline Scientists: Junjing Deng, Jeff Klug and others
Compression and reconstructions: Sheng Di, Tekin Bicer

Ptychographic experiment: reconstruction on (sz) decompressed diffraction patterns.

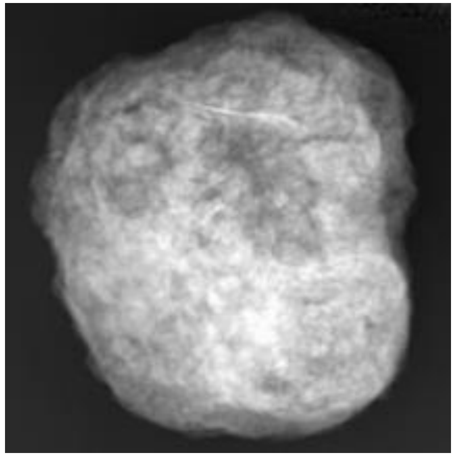
Reconstruction parameters: Iter=300; Alg.:Conjugate Gradient (Tike)



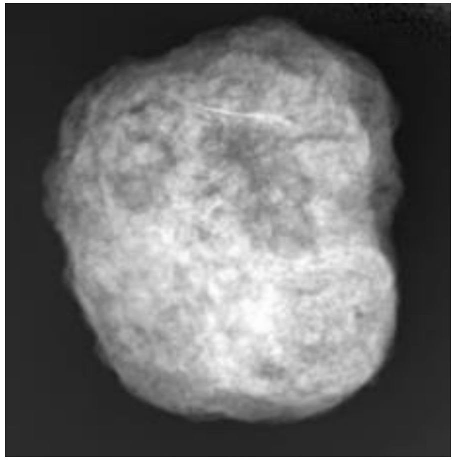
(de)compressed
diff. patterns

Ptycho.
recon.

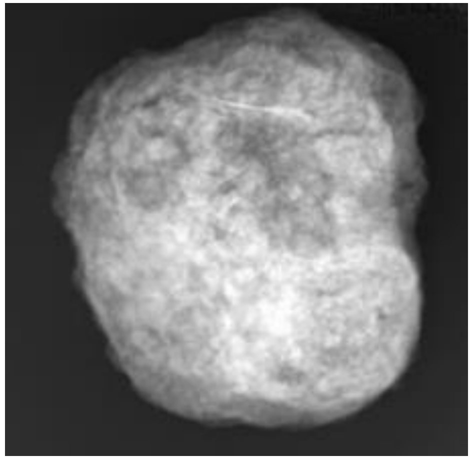
Original
No (de)compression



Spatial error bound: 4
Compression ratio: **97**
SSIM: >0.96



Temporal error bound: 8
Compression ratio: **245**
SSIM: >0.94



Conclusion

Lossy Compression for scientific data:

- Very popular topic among application teams
- SZ is the only customizable compressor
- **… designed to enable science preservation**
- Can tune compression ratio, speed and accuracy according to specific constraints
- Tested on many different applications and experiments
- Generic SZ good enough for Ptychography
- Specific RiobinSZ needed for Crystalography
- Open-source, production quality, integrated in HDF5 and other I/O libs (Adios, NetCDF)

Argonne NATIONAL LABORATORY

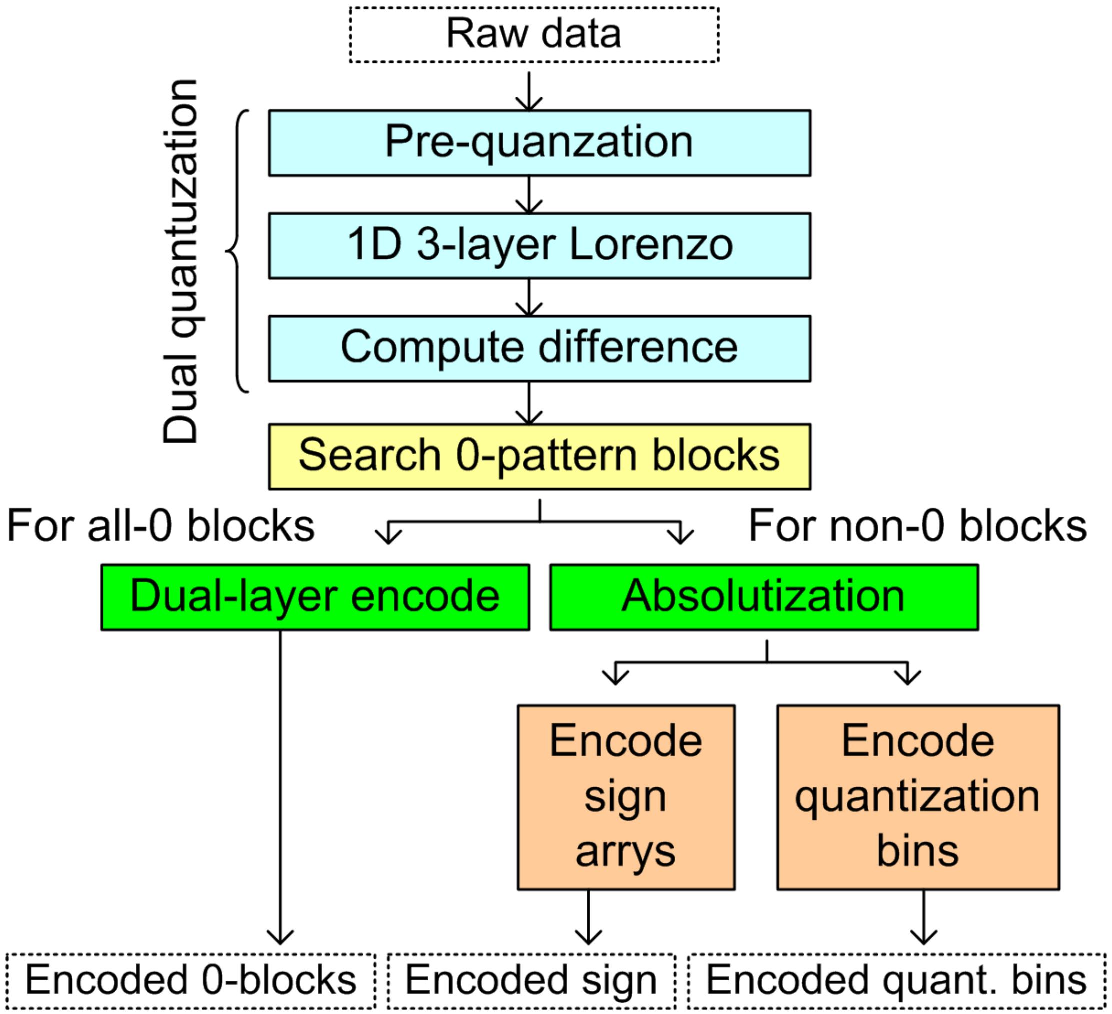
Thanks

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.

Argonne
NATIONAL LABORATORY

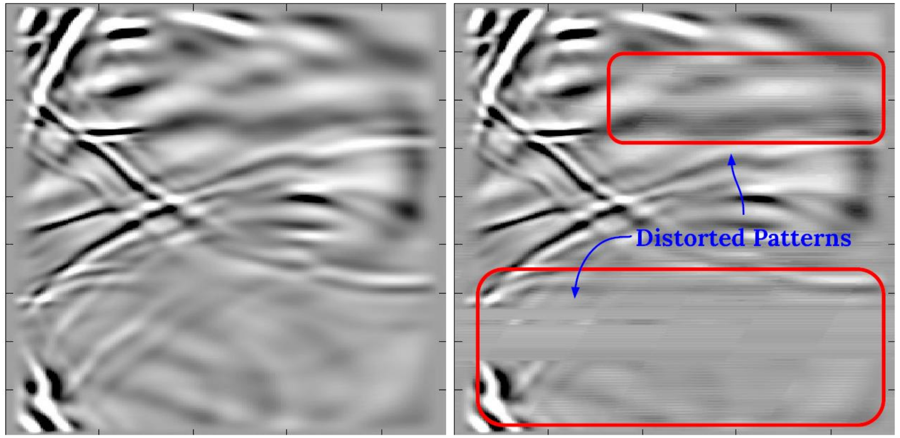
SZp: Design Principle

- **Pre-quantization:** raw data → integers (based on error bound)
- **1D 3-layer Lorenzo:** Data prediction based on only three previous values
- **Compute difference** between predicted value and raw data value
- **Search 0-pattern blocks:** search all-0 blocks (block size = 16)
- **Encode quantization bins:** simplified bit-shuffle-similar encoding



SZp: High Speed Compression

- We have implemented the CPU version to verify SZp's compression ratio and GPU version for speed.
- Compression quality: SZp has much higher quality than SZx on seismic pressure datasets. SZx may have obvious artifacts at high-compression cases (i.e., error bound is relatively high), while SZp has no such issue.



(a) Original Data (b) SZx Reconstructed Data

REL=1E-2,
CR=5

CR	1E-2	1E-3	1E-4	1E-5
SZx	6.1	5.07	4.1	3.5
SZp	19.4	12.3	8.5	6.3

- Compression speed: cuSZp significantly outperforms the BitComp on CUDA A100 when including kernel launch cost: 400GB/s vs. 200GB/s.

More Lossy Compressors

ZFP (LLNL): Transform (DCT)
ECP ZFP
Overpreserves data, lower Compression ratio compared to SZ, Better speed.

SPERR (NCAR): Wavelet
Works well on wave propagation problem (Climate, Seismic)

MGARD (ORNL)
ECP CODAR
Multigrid adaptive reduction
MGARD controls the compression errors in quantities of interest (Q): Linear expression of the error

Largest Compression Ratio For Each Compressor that Satisfies Each Pinard et al (2020) Requirements

Compressor	Pearson R^2	Spatial Error	KS-test
SZ_Interp	93	93	**21**
SZ (regression)	14.34	14.34	**14.34**
ZFP	5.45	5.45	**2.36**
MGARD	27.1	4.69	X
MGARDx	14.7	6.49	X
TThresh	16.1	16.1	**2.98**
BitGrooming	1.51	1.51	**1.51**
Digit Rounding	1.86	1.86	**1.86**
FPZip	1.95	1.95	**1.95**
NDZip	1.64	1.64	**1.64**
Zstd	1.35	1.35	**1.35**

More Lossy Compressors

TTRESH (LLNL):

HoSVD (Tucker Decomposition)
Quantize the Core tensor
Very high compression ratio
Tendency to blur the overall data (loose details)
1 or 2 orders of magnitude slower than SZ or ZFP

Autoencoders

Overall architecture of convolutional autoencoder (A. Glaws, R. King, and M. Sprague, "Deep learning for in situ data compression of large turbulent flow simulations," Physical Review Fluids, vol. 5, no. 11, p. 114602, 2020.)

12 residual blocks for feature extraction + 3 compression layers

Largest Compression Ratio For Each Compressor that Satisfies Each Pinard et al (2020) Requirements

Compressor	Pearson R^2	Spatial Error	KS-test
SZ_Interp	93	93	**21**
SZ (regression)	14.34	14.34	**14.34**
ZFP	5.45	5.45	**2.36**
MGARD	27.1	4.69	X
MGARDx	14.7	6.49	X
TThresh	16.1	16.1	**2.98**
BitGrooming	1.51	1.51	**1.51**
Digit Rounding	1.86	1.86	**1.86**
FPZip	1.95	1.95	**1.95**
NDZip	1.64	1.64	**1.64**
Zstd	1.35	1.35	**1.35**

Significant Smoothing

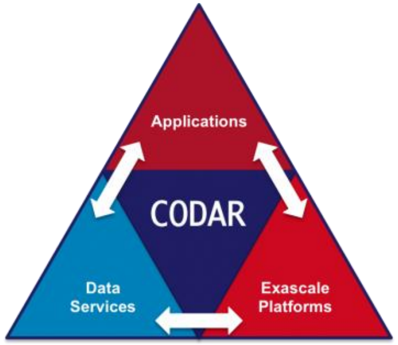
Argonne NATIONAL LABORATORY

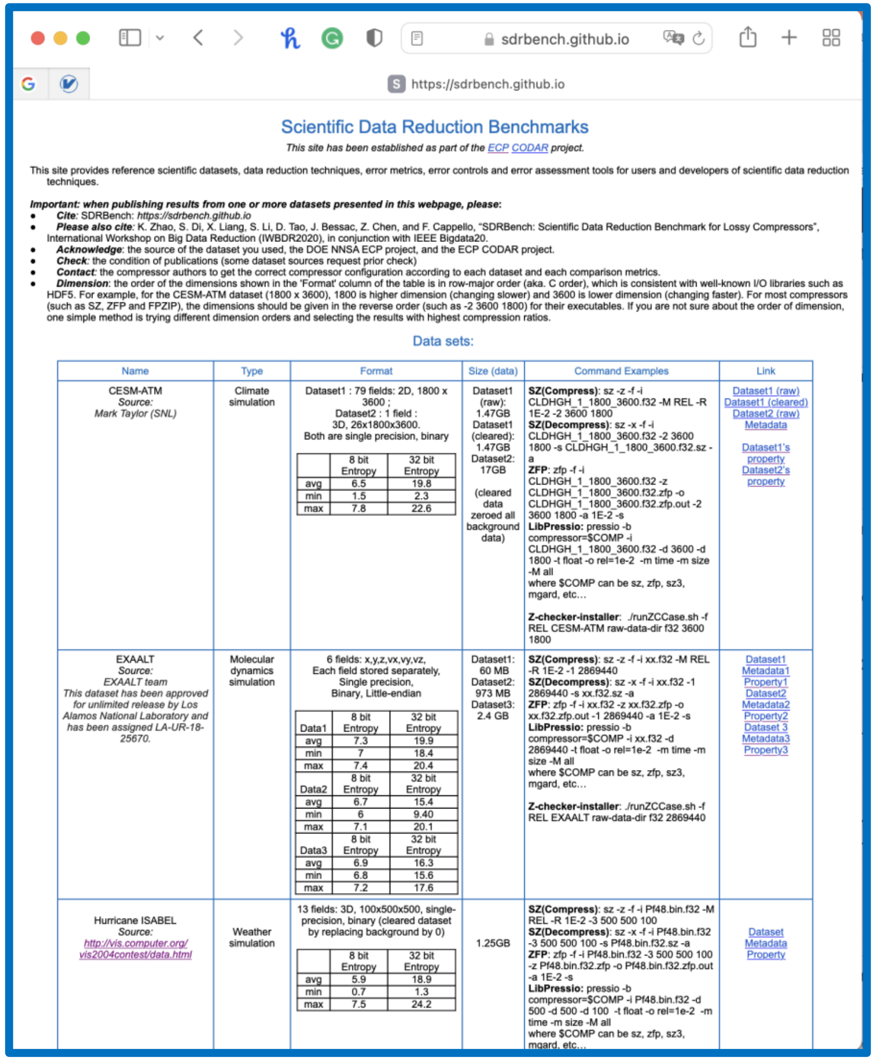
Methodologies

https://sdrbench.github.io/

https://github.com/robertu94/libpressio

https://github.com/CODARcode/Z-checker





sdrbench.github.io

https://sdrbench.github.io

Scientific Data Reduction Benchmarks

This site has been established as part of the ECP CODAR project.

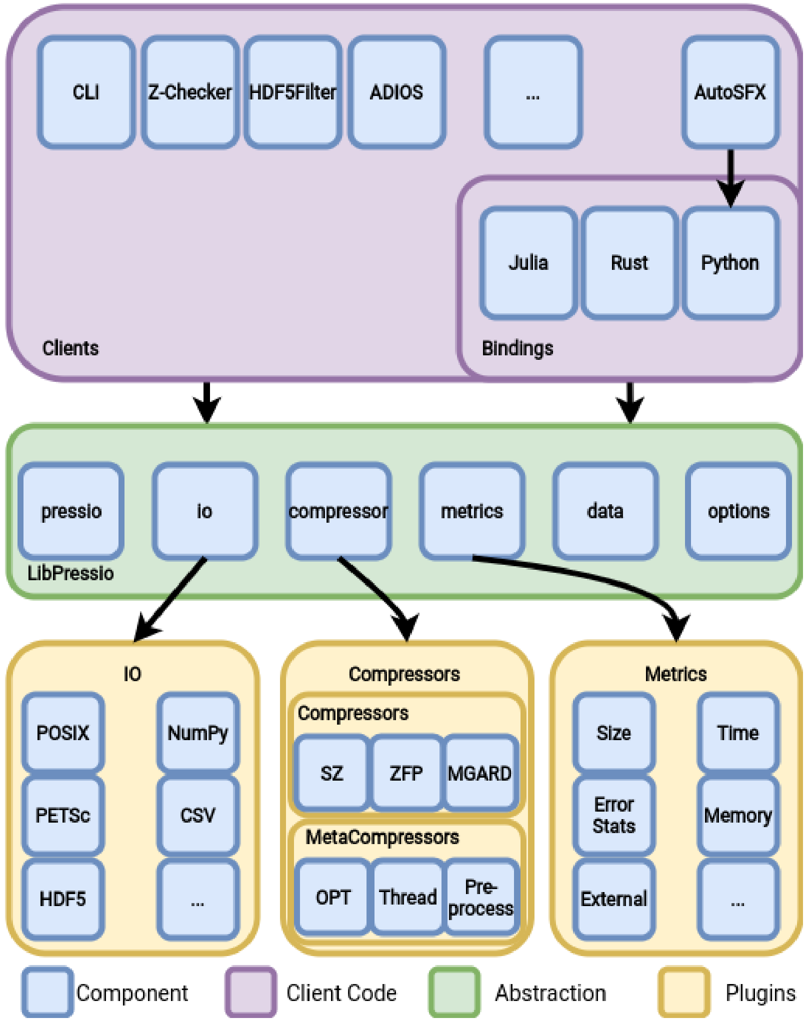
This site provides reference scientific datasets, data reduction techniques, error metrics, error controls and error assessment tools for users and developers of scientific data reduction techniques.

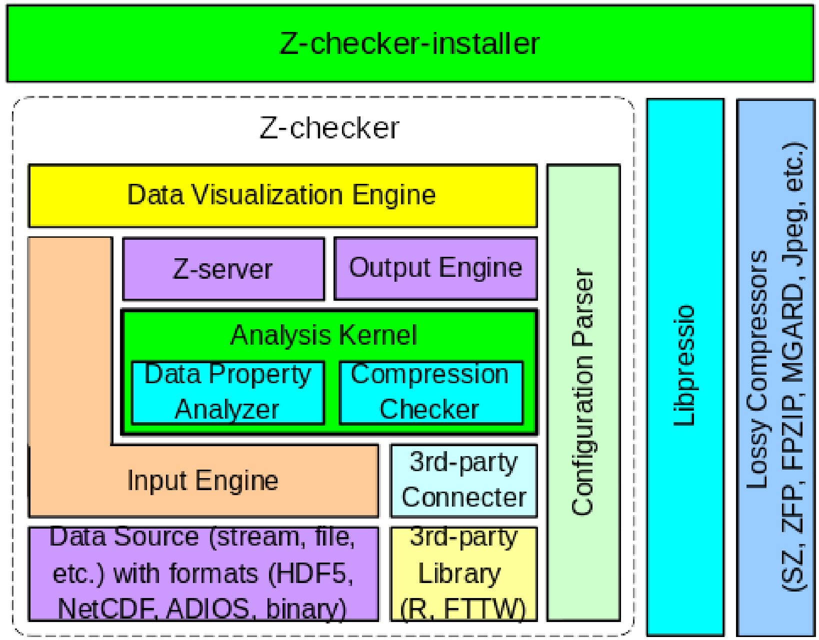
Important: when publishing results from one or more datasets presented in this webpage, please:

- ***Cite***: SDRBench: *https://sdrbench.github.io*
- ***Please also cite***: K. Zhao, S. Di, X. Liang, S. Li, D. Tao, J. Bessac, Z. Chen, and F. Cappello, "SDRBench: Scientific Data Reduction Benchmark for Lossy Compressors", International Workshop on Big Data Reduction (IWBDR2020), in conjunction with IEEE Bigdata20.
- ***Acknowledge***: the source of the dataset you used, the DOE NNSA ECP project, and the ECP CODAR project.
- ***Check***: the condition of publications (some dataset sources request prior check)
- ***Contact***: the compressor authors to get the correct compressor configuration according to each dataset and each comparison metrics.
- ***Dimension***: the order of the dimensions shown in the 'Format' column of the table is in row-major order (aka. C order), which is consistent with well-known I/O libraries such as HDF5. For example, for the CESM-ATM dataset (1800 x 3600), 1800 is higher dimension (changing slower) and 3600 is lower dimension (changing faster). For most compressors (such as SZ, ZFP and FPZIP), the dimensions should be given in the reverse order (such as -2 3600 1800) for their executables. If you are not sure about the order of dimension, one simple method is trying different dimension orders and selecting the results with highest compression ratios.

Data sets:

Name	Type	Format	Size (data)	Command Examples	Link
CESM-ATM *Source:* *Mark Taylor (SNL)*	Climate simulation	Dataset1 : 79 fields: 2D, 1800 x 3600 ; Dataset2 : 1 field : 3D, 26x1800x3600. Both are single precision, binary (8 bit Entropy / 32 bit Entropy) avg 6.5 / 19.8 min 1.5 / 2.3 max 7.8 / 22.6	Dataset1 (raw): 1.47GB Dataset1 (cleared): 1.47GB Dataset2: 17GB (cleared data zeroed all background data)	**SZ(Compress)**: sz -z -f -i CLDHGH_1_1800_3600.f32 -M REL -R 1E-2 -2 3600 1800 **SZ(Decompress)**: sz -x -f -i CLDHGH_1_1800_3600.f32 -2 3600 1800 -s CLDHGH_1_1800_3600.f32.sz -a **ZFP**: zfp -f -i CLDHGH_1_1800_3600.f32 -z CLDHGH_1_1800_3600.f32.zfp -o CLDHGH_1_1800_3600.f32.zfp.out -2 3600 1800 -a 1E-2 -s **LibPressio:** pressio -b compressor=$COMP -i CLDHGH_1_1800_3600.f32 -d 3600 -d 1800 -t float -o rel=1e-2 -m time -m size -M all where $COMP can be sz, zfp, sz3, mgard, etc… **Z-checker-installer**: ./runZCCase.sh -f REL CESM-ATM raw-data-dir f32 3600 1800	Dataset1 (raw) Dataset1 (cleared) Dataset2 (raw) Metadata Dataset1's property Dataset2's property
EXAALT *Source:* *EXAALT team* *This dataset has been approved for unlimited release by Los Alamos National Laboratory and has been assigned LA-UR-18-25670.*	Molecular dynamics simulation	6 fields: x,y,z,vx,vy,vz, Each field stored separately, Single precision, Binary, Little-endian Data1 (8 bit Entropy / 32 bit Entropy) avg 7.3 / 19.9 min 7 / 18.4 max 7.4 / 20.4 Data2 (8 bit Entropy / 32 bit Entropy) avg 6.7 / 15.4 min 6 / 9.40 max 7.1 / 20.1 Data3 (8 bit Entropy / 32 bit Entropy) avg 6.9 / 16.3 min 6.8 / 15.6 max 7.2 / 17.6	Dataset1: 60 MB Dataset2: 973 MB Dataset3: 2.4 GB	**SZ(Compress)**: sz -z -f -i xx.f32 -M REL -R 1E-2 -1 2869440 **SZ(Decompress)**: sz -x -f -i xx.f32 -1 2869440 -s xx.f32.sz -a **ZFP**: zfp -f -i xx.f32 -z xx.f32.zfp -o xx.f32.zfp.out -1 2869440 -a 1E-2 -s **LibPressio:** pressio -b compressor=$COMP -i xx.f32 -d 2869440 -t float -o rel=1e-2 -m time -m size -M all where $COMP can be sz, zfp, sz3, mgard, etc… **Z-checker-installer**: ./runZCCase.sh -f REL EXAALT raw-data-dir f32 2869440	Dataset1 Metadata1 Property1 Dataset2 Metadata2 Property2 Dataset 3 Metadata3 Property3
Hurricane ISABEL *Source:* *http://vis.computer.org/vis2004contest/data.html*	Weather simulation	13 fields: 3D, 100x500x500, single-precision, binary (cleared dataset by replacing background by 0) (8 bit Entropy / 32 bit Entropy) avg 5.9 / 18.9 min 0.7 / 1.3 max 7.5 / 24.2	1.25GB	**SZ(Compress)**: sz -z -f -i Pf48.bin.f32 -M REL -R 1E-2 -3 500 500 100 **SZ(Decompress)**: sz -x -f -i Pf48.bin.f32 -3 500 500 100 -s Pf48.bin.f32.sz -a **ZFP**: zfp -f -i Pf48.bin.f32 -3 500 500 100 -z Pf48.bin.f32.zfp -o Pf48.bin.f32.zfp.out -a 1E-2 -s **LibPressio:** pressio -b compressor=$COMP -i Pf48.bin.f32 -d 500 -d 500 -d 100 -t float -o rel=1e-2 -m time -m size -M all where $COMP can be sz, zfp, sz3, mgard, etc…	Dataset Metadata Property

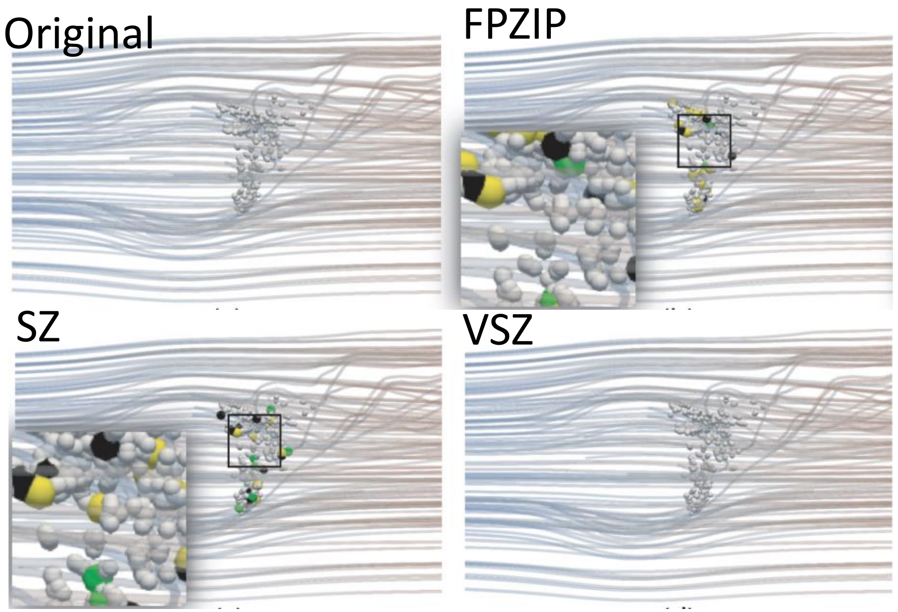




VSZ

Features to preserve are mathematically formalized and integrated into compressor error controls

- VSZ (SZ-Critical points): Preserves Critical points in 2D, 3D piecewise linear vector fields (Important in flow visualization, keep each critical point in its original cell, retain each critical point type).
  Compute error bound on each data point.



X. Liang *et al*., Toward Feature-Preserving 2D and 3D Vector Field Compression, *IEEE Pacific Visualization Symposium (PacificVis)*, 2020

Provides excellent compression performance and feature preservation.

Limitations:

- Expressing feature mathematically could be too complex.
- Requires specific compression algorithm designs for each feature to preserve.
- Preservation of combination of features has not been addressed.