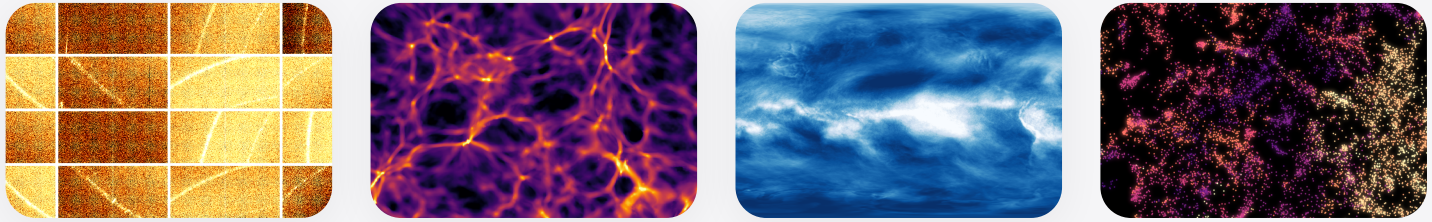
TECHNICAL PAPERS | BEST PAPER FINALIST

FSZ

FSZ: Breaking the Prediction-Throughput Trade-Off in GPU Lossy Compression

Jiajun Huang, University of South Florida
jiajunhuang@usf.edu



SC26 Chicago, IL hpc unites.

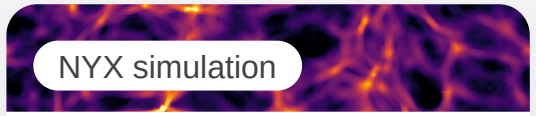
INTRODUCTION

Science and AI Produce Data Faster Than It Can Be Stored



> 1 TB/s

X-ray laser: LCLS-II detector data by the end of the decade, up from about 250 GB/s in 2023 [Thayer et al., CHEP'23]



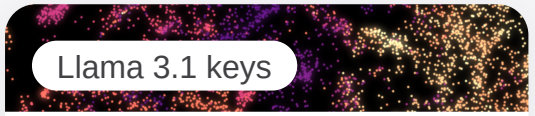
> 100 PB

Cosmology: written by one 4-trillion-particle run on Frontier in slightly more than one week [Frontiere et al., SC'25]



1.7 PB

Climate: compressed output of about 1,000 CESM2 runs for CMIP6 [Danabasoglu, WGCM'20]

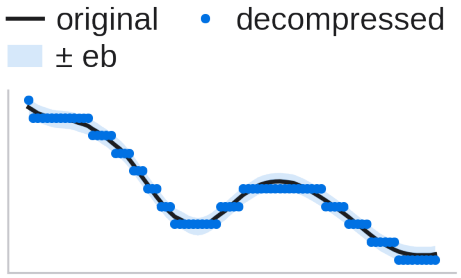


40 GiB

AI: KV cache of Llama-3.1-70B for one 128K-token prompt, 320 KiB per token [Grattafiori et al., arXiv'24]

Error-bounded lossy compression

Compression ratios of 10 to 100, with every value within a bound *eb* set by the scientist.



Data movement is the bottleneck

Storage: about 0.5 GB/s per node when all 9,000 Frontier nodes write [Frontiere et al., SC'25]

Network: 50 GB/s per 400 Gb/s link, against 8 TB/s of B200 memory bandwidth [NVIDIA, DGX B200 datasheet]

Therefore, the data must be compressed on the GPU, before they leave GPU memory.

SC26 Chicago, IL hpc unites.

Use Cases for Fast GPU Compression



Stream instrument data

Example
LCLS-II employs error-bounded lossy compression in its real-time data reduction [Thayer et al., CHEP'23].



Store simulation output and checkpoints

Example
Frontier-E writes 150 to 180 TB per checkpoint [Frontiere et al., SC'25].



Transfer data between GPUs

Example
Compression-accelerated collectives gZCCL and ghZCCL [Huang et al., ICS'24, ICS'25].



Train and serve AI models

Example
COMET reduces CNN training memory by up to 13.5× with SZ [Jin et al., PVLDB'22]. LLM serving moves the KV cache from prefill to decode GPUs [Zhong et al., OSDI'24].

Every use case requires **all three at once**

- **Hundreds of GB/s on the GPU**
- **A high compression ratio**
- **A guaranteed error bound**

INTRODUCTION

The Prediction-Throughput Trade-Off

Fast GPU compressors

cuSZp [Huang et al., SC'23, SC'24, SC'25],
FZ-GPU [Zhang et al., HPDC'23]

- Hundreds of GB/s on the GPU
- Fixed first-order prediction on independent blocks
- Compression ratio limited by the predictor

High-ratio GPU compressors

cuSZ-Hi [Wu et al., SC'25], PFPL
[Fallin et al., IPDPS'25]

- Richer prediction and lossless encoding stages
- Higher compression ratio
- Substantially lower GPU throughput

Common assumption: *richer prediction inevitably reduces GPU compressor throughput.*

FSZ: richer prediction inside a fast single-kernel compressor

up to **2.92×**
the compression ratio
of cuSZp-O [Huang et al., SC'24]

676 / **785** GB/s
compression / decompression,
highest of all evaluated compressors

DESIGN

Three Challenges for Fast GPU Compressors

Partition the data into blocks of 32 values

Store differences from the left neighbor

Set the bit width by the largest difference

For example, the state-of-the-art cuSZp [Huang et al., SC'23, SC'24, SC'25] follows this design, in plain mode (cuSZp-P) and outlier mode (cuSZp-O).

1 Boundary overhead of independent blocks

Differences in one block of 32 values

1000	1	1	…	1

10 bits per value: **44 bytes** instead of **8**

The first value of each block has no left neighbor, so its difference is the full value, and one large difference sets the bit width of all 32 values.

2 Fixed prediction cannot adapt to the data



flat pieces smooth trend large offset

real CESM-ATM rows

Fast compressors apply first-order prediction everywhere, while the pattern changes between fields and within each field: flat pieces, smooth trends, or a large offset.

3 Richer prediction must maintain the throughput

Compression throughput

1 read per value

+1 read per value **30 to 50% lower**

Fast compressors are memory-bound, so testing several predictors with one read of the data per predictor would cut the throughput.

SC26 Chicago, IL hpc unites.

DESIGN

Key Insight: Compute Headroom in Memory-Bound Passes

Read 1
Determine the bit width of each block

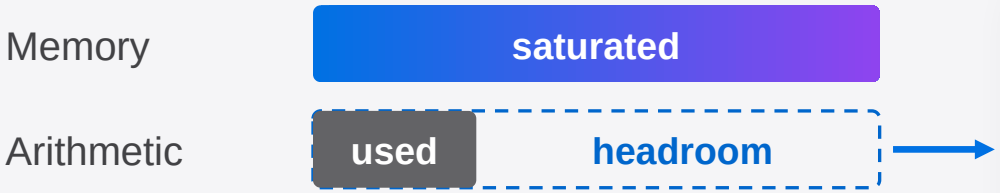
→

Offsets
Sum the block sizes to locate each block

→

Read 2
Write each block at its offset

During read 1 (memory-bound):



Schematic illustration

→

FSZ converts this headroom into compression ratio

Richer prediction operates on values already in registers, so FSZ maintains the memory traffic of a fast compressor.

- Every fast single-kernel GPU compressor reads the data twice: the second read requires the offsets computed after the first.
- Read 1 is memory-bound: memory bandwidth determines its execution time, and arithmetic on values in registers adds almost no time.

SC26 Chicago, IL hpc unites.

DESIGN

FSZ: Three Designs in a Single GPU Kernel

1 Carry prediction state across blocks

Independent blocks: 8 large residuals



FSZ: 1 large residual per 8 blocks



One thread carries the last value of each block into the next block, within its own registers.

2.00×

the ratio of cuSZp-O [Huang et al., SC'24] on a turbulence field (Miranda density)

2 Select the best predictor per tile



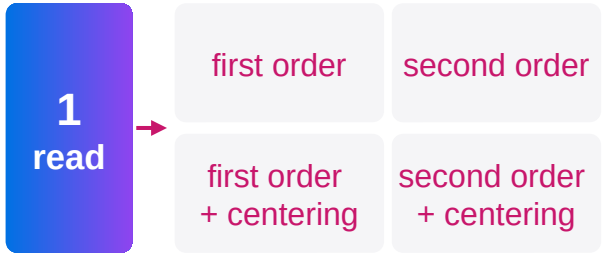
flat pieces	smooth trend	large offset
↓	↓	↓
first order	second order	centering

Each tile of 256 values selects first- or second-order prediction, each optionally with centering.

2.92×

the ratio of cuSZp-O on a climate field (CESM-ATM Z3)

3 Evaluate four variants in one read



A constant offset cancels in differences, so one read yields the exact size of all four variants.

2.8×

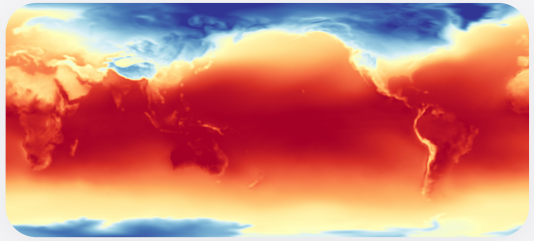
the throughput of one data read per variant, at the same ratio

All three together: at every error bound, FSZ attains a higher average compression ratio and a higher throughput than every evaluated compressor

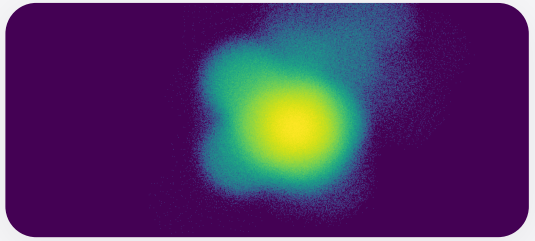
SC26 Chicago, IL | hpc unites.

EVALUATION

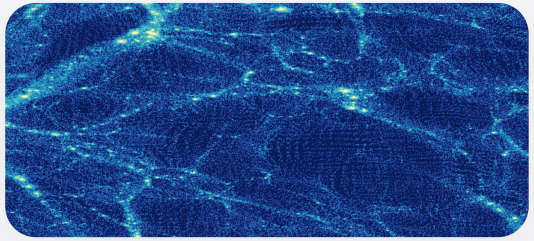
Evaluated on 8 Science Domains, 84 Real Application Fields



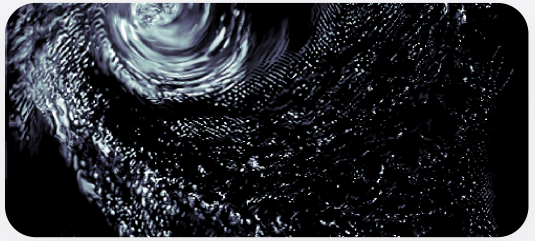
Climate
CESM-ATM, 33 fields



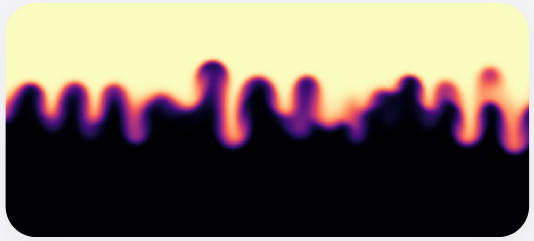
Materials science
EXAALT, 6 fields



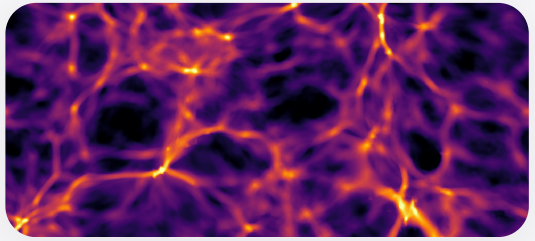
Cosmology
HACC, 6 fields



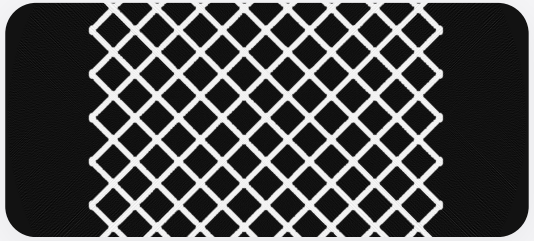
Hurricanes
Hurricane, 13 fields



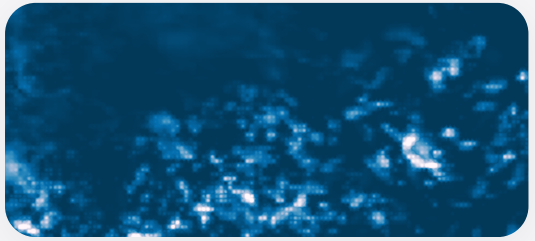
Turbulence
Miranda, 7 fields



Cosmology
NYX, 6 fields



Advanced manufacturing
Truss, simulated CT scan, 1 field



Weather
SCALE, 12 fields

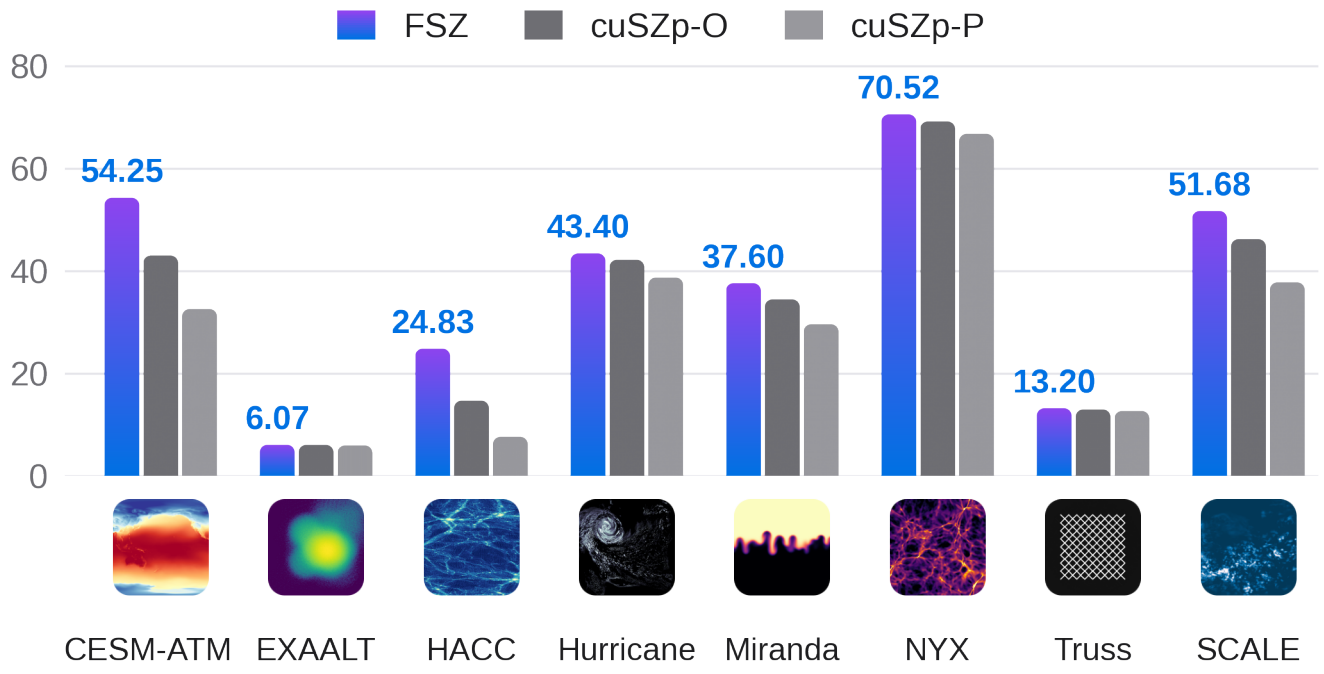
Data: SDRBench [Zhao et al., BigData'20] and Truss [Klacansky et al., PacificVis'22], at relative error bounds 1E-2, 1E-3 and 1E-4

Baselines: cuSZp-P [Huang et al., SC'23], cuSZp-O [Huang et al., SC'24], FZ-GPU [Zhang et al., HPDC'23] and cuZFP [Lindstrom, TVCG'14], on NVIDIA GH200 and A100 GPUs

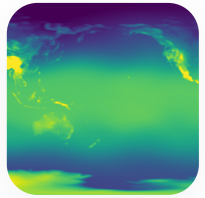
SC26 Chicago, IL hpc unites.

EVALUATION

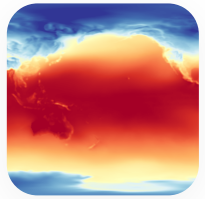
FSZ Attains the Highest Average Compression Ratio



cuSZp-O [Huang et al., SC'24] and cuSZp-P [Huang et al., SC'23].



up to **2.92×**
the ratio of cuSZp-O
on CESM-ATM Z3



up to **10.95×**
the ratio of cuSZp-P
on CESM-ATM T

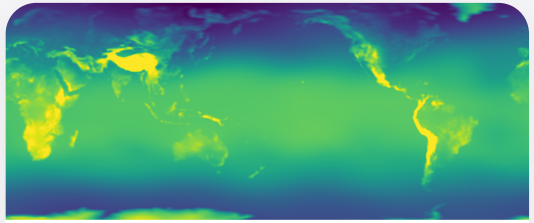
Highest average on every dataset

FSZ attains the highest average ratio on all 8 datasets in this chart.

Identical accuracy: FSZ and cuSZp reconstruct identical values at the same error bound, so FSZ attains the higher ratio at identical accuracy.

EVALUATION

Each Design Benefits a Different Science Domain



Climate (CESM-ATM)
Geopotential height (Z3)

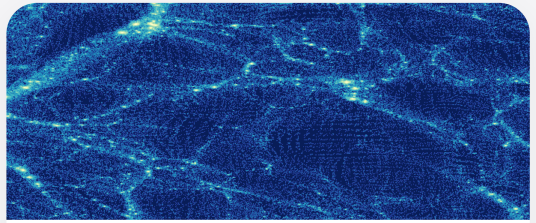
Smooth vertical gradient

Captured by

second order

2.92×

the ratio of cuSZp-O



Cosmology (HACC)
Particle positions (yy)

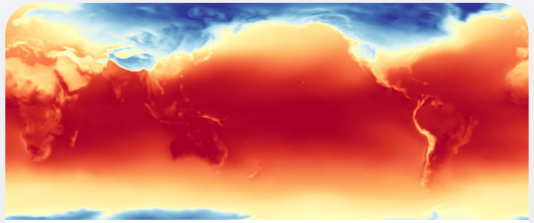
Near-linear distribution

Captured by

second order

2.61×

the ratio of cuSZp-O



Climate (CESM-ATM)
Temperature (T)

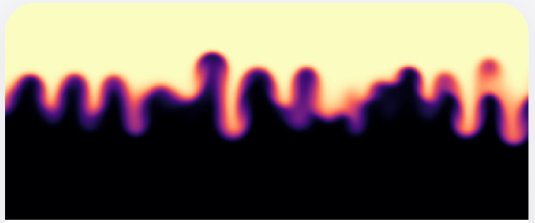
Large offset (Kelvin)

Captured by

centering

2.20×

the ratio of cuSZp-O



Turbulence (Miranda)
Density

Strong spatial structure

Captured by

cross-block state

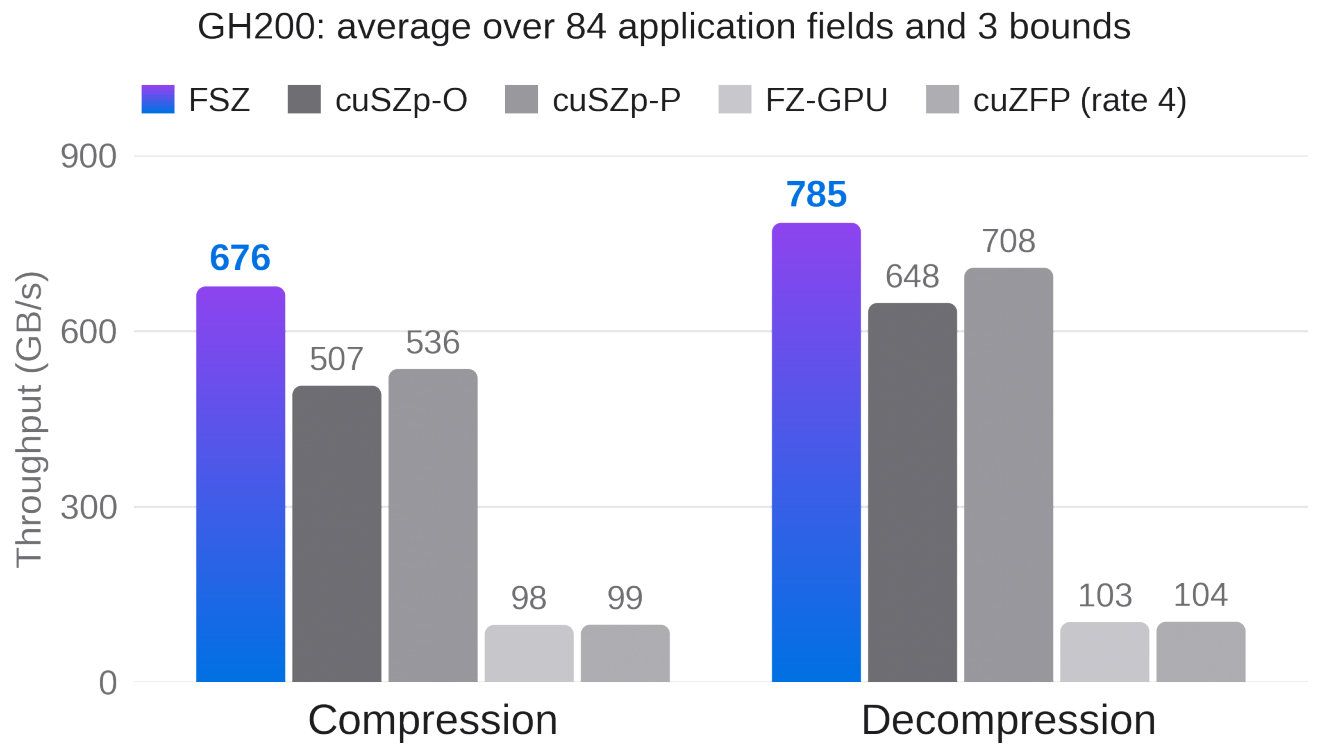
2.00×

the ratio of cuSZp-O

Per-field ratio of FSZ over cuSZp-O [Huang et al., SC'24] at REL 1E-2 on SDRBench fields [Zhao et al., BigData'20], with the pattern and design identified in the paper.

EVALUATION

FSZ Attains the Highest Throughput



cuSZp-P [Huang et al., SC'23], cuSZp-O [Huang et al., SC'24], FZ-GPU [Zhang et al., HPDC'23] and cuZFP [Lindstrom, TVCG'14] at fixed rate 4, its fastest setting.

1.26× / 1.33×

the compression throughput of cuSZp-P / cuSZp-O

1.11× / 1.21×

the decompression throughput of cuSZp-P / cuSZp-O

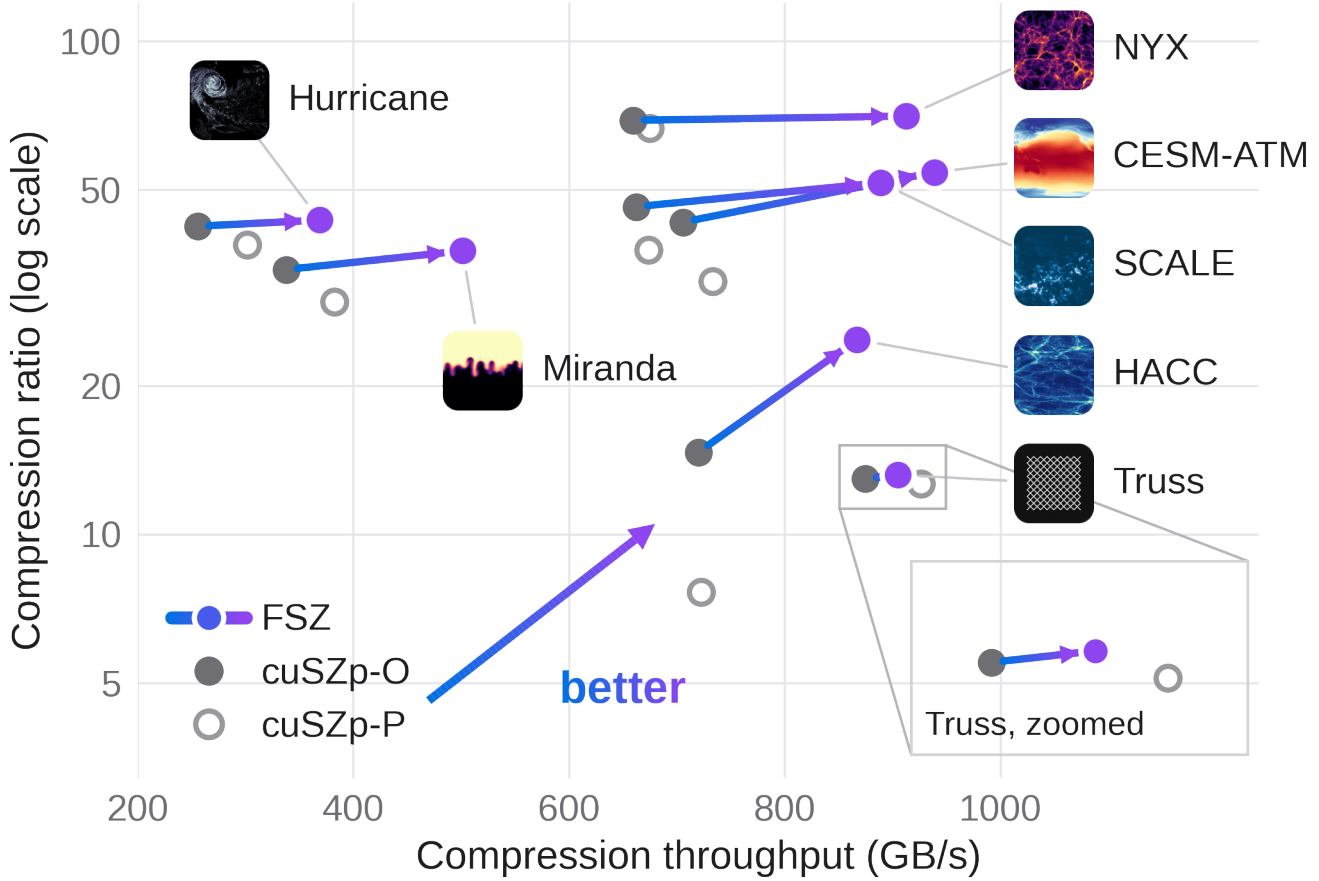
Why FSZ is faster

- One read evaluates four variants
- Follows the GPU execution model
- Higher ratio accelerates I/O

Per field at REL 1E-4: up to **1.65×** (compression) and **2.31×** (decompression) the throughput of cuSZp-O.

EVALUATION

FSZ Improves Ratio and Throughput on Every Dataset



1.02× to 1.69×

the compression ratio of cuSZp-O [Huang et al., SC'24]

1.03× to 1.48×

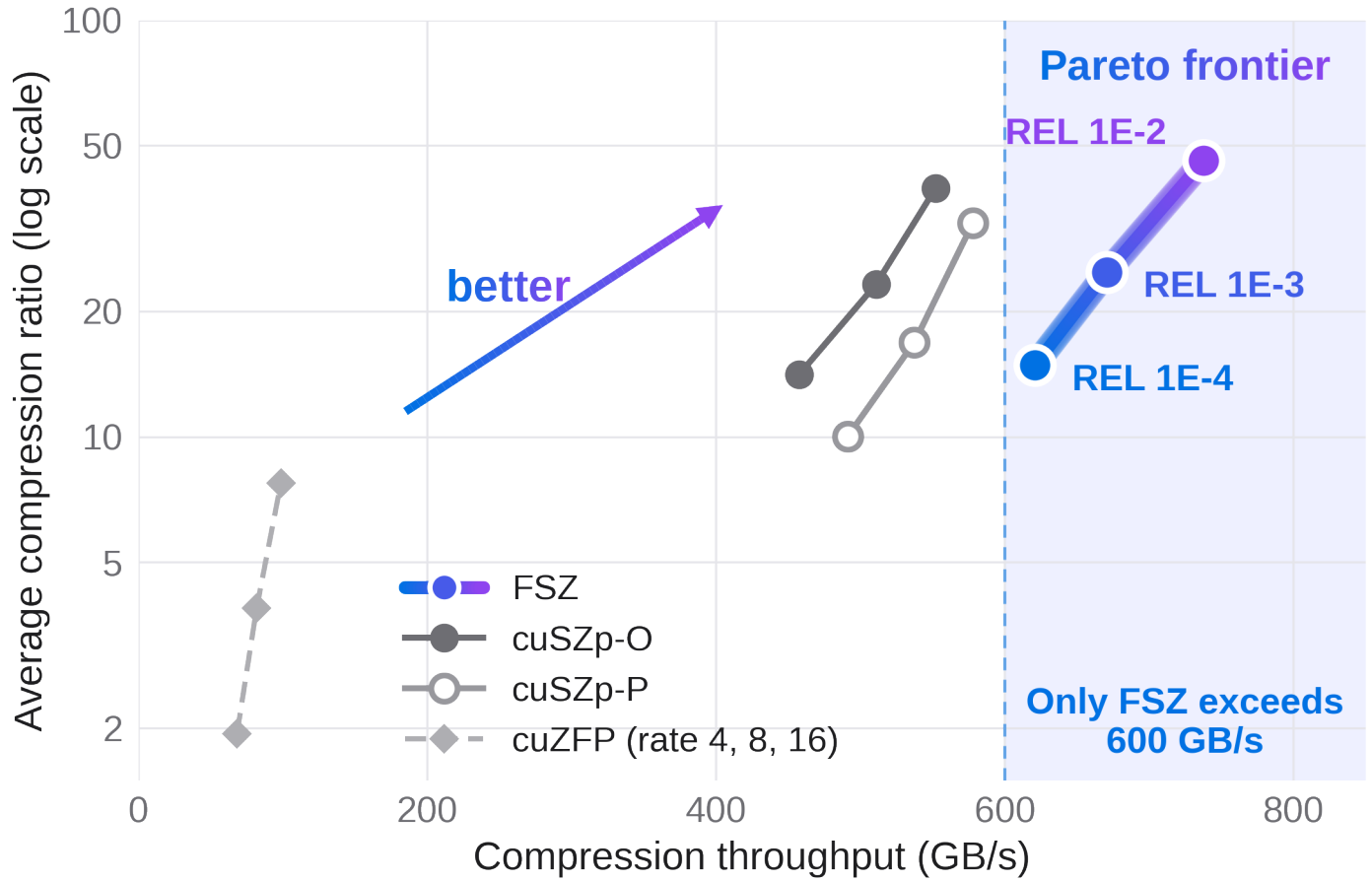
the compression throughput of cuSZp-O

Higher ratio and higher throughput, on every dataset.

Per-dataset averages at REL 1E-2. Hollow circles: cuSZp-P [Huang et al., SC'23]. EXAALT (11 MB fields) lies outside the plotted range, at 1.41× the throughput of cuSZp-O.

EVALUATION

FSZ Stands at the Pareto Frontier



At every error bound

FSZ attains a higher average ratio and a higher throughput than every evaluated compressor.

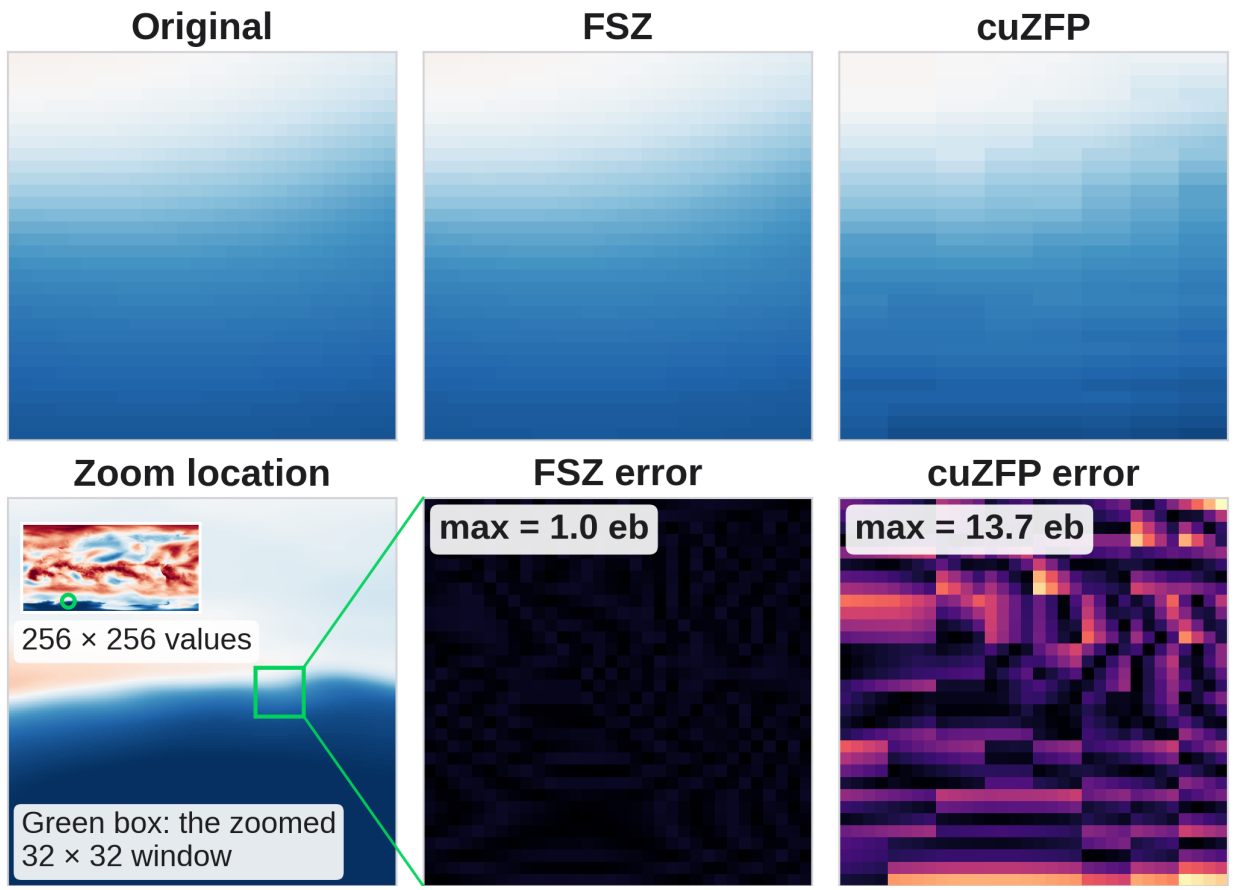
How to read it

Each curve is one compressor, from REL 1E-4 at the bottom to REL 1E-2 at the top. The FSZ curve lies above and to the right of all others.

Averages over 84 application fields. Baselines: cuSZp-O [Huang et al., SC'24], cuSZp-P [Huang et al., SC'23] and cuZFP [Lindstrom, TVCG'14] at rates 4, 8 and 16.

EVALUATION

Higher Data Quality at the Same Size



14.88 vs **14.86**

compression ratio of FSZ and cuZFP
[Lindstrom, TVCG'14] (fixed rate 2)

61.96 vs **53.27** dB

PSNR over the full field:
FSZ is 8.7 dB higher

0% vs **23.0%**

points beyond the error bound, over all 168.5M points

Best rate-distortion

among all evaluated
error-bounded compressors

EVALUATION

FSZ Accelerates Three Applications End to End



GPU memory → Lustre



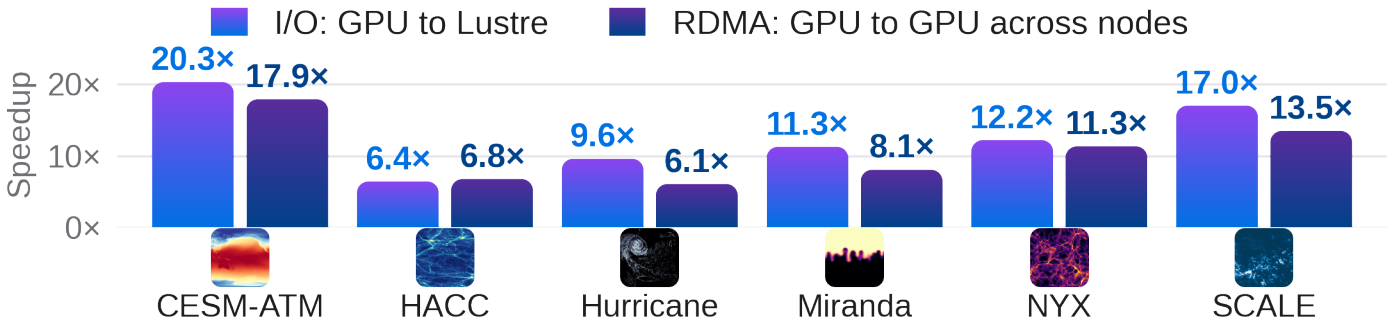
node 1 → node 2

I/O: GPU to file system

12.0× faster writes

Communication: GPU to GPU

9.8× faster transfers



Every field of six snapshots (1 to 21 GiB) at REL 1E-3 on NVIDIA B200 nodes, geometric mean over the datasets. Uncompressed: 5.2 to 7.7 GB/s to Lustre (write and fsync) and 49 GB/s over 400 Gb/s InfiniBand with GPUDirect RDMA.



prefill GPU → decode GPU

KV cache: LLM serving

3.4× faster transfer

of the 40 GiB KV cache of Llama-3.1-70B, one 128K-token prompt, two B200 nodes

RULER 128K, 25/25 uncompressed and 25/25 with FSZ [Hsieh et al., COLM'24]

Llama-3.1-8B, 10.7K to 31.4K tokens: 2.4× faster, with LongBench [Bai et al., ACL'24] scores matching

Llama 3.1 Instruct [Grattafiori et al., arXiv'24] at REL 1E-2, prefill and decode on two B200 nodes [Zhong et al., OSDI'24]. Ratio over bf16: 3.4 for 70B, 3.3 for 8B.

SC26 Chicago, IL hpc unites.

CONCLUSION

FSZ Breaks the Prediction-Throughput Trade-Off

Higher throughput
676 GB/s compression and 785 GB/s decompression on GH200, the highest of all evaluated compressors

Higher compression ratio
Up to 2.92× and 10.95× the compression ratio of cuSZp-O and cuSZp-P [Huang et al., SC'24, SC'23]

Identical accuracy
The same values as cuSZp at the same error bound, and 8.7 dB higher PSNR than cuZFP [Lindstrom, TVCG'14] at the same size

Faster science and AI applications
12.0× faster writes to the file system and 9.8× faster GPU-to-GPU transfers of science data, and up to 3.4× faster KV-cache transfers in LLM serving

Three designs in a single GPU kernel

1. Carry prediction state across blocks
2. Select the best predictor per tile
3. Evaluate four variants in one read

Richer prediction in the compute headroom of a memory-bound pass

Next: apply the same co-design to other hardware, such as FPGAs and AI accelerators (e.g., Cerebras WSE)

CONCLUSION

FSZ Is Available Today

Command line

```
$ fsz -i velocity.f32 -eb rel 1e-3
```

The tool reports the compression ratio, the throughput and the maximum error. The options -z and -x write and read .fsz files.

Python

```
import fsz
blob = fsz.compress(arr, eb=1e-3, mode="rel")
out = fsz.decompress(blob)
```

github.com/szcompressor/FSZ

Release 1.0.0, BSD 3-Clause license

Part of the SZ compressor family

Released through the SZ project and directly available to SZ users

Five interfaces

Command line, C, C++, Python and Fortran, for float32 and float64 data

Automatic adaptation

FSZ adapts to the data automatically, and the tool verifies the error bound on every run

Portable across NVIDIA GPUs

On B200: up to 1.4 TB/s compression and 1.7 TB/s decompression

Reproducible artifact

doi.org/10.5281/zenodo.21781794

SC26 Chicago, IL hpc unites.

References

Applications

J. Thayer, Z. Chen, R. Claus, et al. Massive Scale Data Analytics at LCLS-II. EPJ Web of Conferences 295 (CHEP 2023), 2024.

N. Frontiere, J. D. Emberson, M. Buehlmann, et al. Cosmological Hydrodynamics at Exascale: A Trillion-Particle Leap in Capability. SC 2025.

G. Danabasoglu. Community Earth System Model (CESM) and CMIP6. WCRP WGCM-23 meeting, 2020.

J. Huang, S. Di, X. Yu, et al. gZCCL: Compression-Accelerated Collective Communication Framework for GPU Clusters. ICS 2024.

J. Huang, S. Di, Y. Huang, et al. ghZCCL: Advancing GPU-aware Collective Communications with Homomorphic Compression. ICS 2025.

S. Jin, C. Zhang, X. Jiang, et al. COMET: A Novel Memory-Efficient Deep Learning Training Framework by Using Error-Bounded Lossy Compression. PVLDB 2022.

Y. Zhong, S. Liu, J. Chen, et al. DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving. OSDI 2024.

A. Grattafiori, A. Dubey, A. Jauhri, et al. The Llama 3 Herd of Models. arXiv:2407.21783, 2024.

Y. Bai, X. Lv, J. Zhang, et al. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. ACL 2024.

C.-P. Hsieh, S. Sun, S. Kriman, et al. RULER: What's the Real Context Size of Your Long-Context Language Models? COLM 2024.

Compressors and methods

Y. Huang, S. Di, X. Yu, et al. cuSZp: An Ultra-fast GPU Error-bounded Lossy Compression Framework with Optimized End-to-End Performance. SC 2023.

Y. Huang, S. Di, G. Li, F. Cappello. cuSZp2: A GPU Lossy Compressor with Extreme Throughput and Optimized Compression Ratio. SC 2024.

Y. Huang, S. Di, G. Li, F. Cappello. GPU Lossy Compression for HPC Can Be Versatile and Ultra-Fast. SC 2025.

B. Zhang, J. Tian, S. Di, et al. FZ-GPU: A Fast and High-Ratio Lossy Compressor for Scientific Computing Applications on GPUs. HPDC 2023.

S. Wu, J. Pan, J. Liu, et al. Boosting Scientific Error-Bounded Lossy Compression through Optimized Synergistic Lossy-Lossless Orchestration. SC 2025.

A. Fallin, N. Azami, S. Di, et al. Fast and Effective Lossy Compression on GPUs and CPUs with Guaranteed Error Bounds. IPDPS 2025.

P. Lindstrom. Fixed-Rate Compressed Floating-Point Arrays. IEEE TVCG 2014.

D. Merrill, M. Garland. Single-pass Parallel Prefix Scan with Decoupled Look-back. NVIDIA Technical Report NVR-2016-002, 2016.

Data and hardware

K. Zhao, S. Di, X. Liang, et al. SDRBench: Scientific Data Reduction Benchmark for Lossy Compressors. IEEE BigData 2020.

P. Klacansky, H. Miao, A. Gyulassy, et al. Virtual Inspection of Additively Manufactured Parts. IEEE PacificVis 2022.

NVIDIA. NVIDIA DGX B200 Datasheet. resources.nvidia.com.

SC26 Chicago, IL | hpc unites.

QUESTIONS

Thank You

Jiajun Huang, University of South Florida
jiajunhuang@usf.edu

Supported by startup funding from the University of South Florida.

github.com/szcompressor/FSZ

SC26
Chicago, IL | hpc unites.