



Lossy Compression for Scientific Data

Karshilov

A Tour of Error-Bounded Lossy Compression

科学数据的“洪水”

The Data Deluge in HPC & Instruments

Exascale 时代的存储危机

- ▶ 气候模拟 (SCREAM): 3.25 km 分辨率, **4.5 TB/天**
- ▶ 宇宙学 (Nyx): 16384^3 网格, 单 checkpoint **~10 PB**
- ▶ 光源 (LCLS-II): 面阵探测器 **1 TB/s**
- ▶ 分子动力学: 长时间轨迹, 存储需求随时间线性增长

Application	Data scale	Passive solution	To reduce
Nyx Cosmology simulation	2.8 TB per snapshot	34 GB/S on Hopper@NERSC	10x-100x In need
CESM Climate simulation	20% vs 50% of h/w budget for storage 2013 vs 2017	5h30m to store NSF Blue Waters, I/O at 1TBps	10x In need
HACC Cosmology simulation	20PB per one-trillion particle simulation	use up FS 26 PB for Mira@ANL	10x In need

存储容量 & I/O 带宽成为科学发现的瓶颈

为什么需要有损压缩?

Lossless $\ll$ **Lossy** for **Floating-Point Data**

无损 (Lossless): 浮点科学数据压缩比通常只有 $\sim 2\times$

- ▶ 浮点尾数近似随机 $\Rightarrow$ 熵编码几乎无效

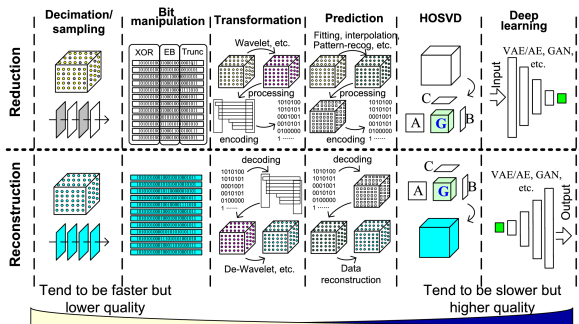
有损 (Lossy): 允许受控误差, 压缩比可达 **10-1000** $\times$

- ▶ 关键概念: **Error Bound**(误差界)
- ▶ **Absolute bound:** $|x_i - \tilde{x}_i| \leq \varepsilon$ / **Relative bound:** $|x_i - \tilde{x}_i|/|x_i| \leq \varepsilon_r$
- ▶ 只要误差可控, 下游分析结果仍然有效

	压缩比	误差控制
Gzip / Zstd (无损)	$\sim 2\times$	0
SZ / ZFP (有损)	$10-100\times$	用户指定
TTHRESH (有损)	$100-1000\times$	L2-norm

压缩模型分类

Six Compression Models



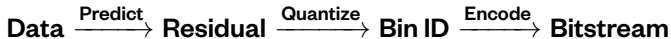
从左到右: 速度递减, 质量递增(大趋势)

1. **Decimation/Filtering** — 采样/降维
2. **Bit Manipulation** — 截断低位, 极快
3. **Transform** — 小波/正交变换
4. **Prediction** — 预测+量化+编码
5. **HOSVD/Tucker** — 张量分解, 超高比
6. **Deep Learning** — AE/VAE

Prediction-based 压缩

SZ 系列 — 最广泛使用的方法

核心 Pipeline:



预测器 (Predictor)

- ▶ Lorenzo: 邻域差分 (d -维)
- ▶ 线性回归 (SZ2)
- ▶ Spline 插值 (SZ3, QoZ)

关键约束

- ▶ 必须用重建值预测(压/解一致性)
- ▶ 扫描顺序可恢复 (RRS Policy)

量化 + 编码

- ▶ 线性量化: bin 宽 = 2ϵ
- ▶ 残差 $\rightarrow$ bin ID (集中在 0 附近)
- ▶ Huffman / Zstd 编码

优缺点

- + 高压缩比 + 严格误差控制
- 变长编码难 GPU 并行

Transform-based 压缩

ZFP & SPERR

ZFP (Lindstrom, LLNL)

- ▶ 4^d 分块正交变换
- ▶ Embedded coding(逐 bit-plane)
- ▶ 支持 fixed-rate / fixed-precision / fixed-accuracy 模式
- ▶ GPU 实现成熟
- + 分块并行友好
- + random access 支持
- pointwise error 控制不如 SZ

SPERR (Li, NCAR)

- ▶ CDF 9/7 小波变换
- ▶ SPECK 编码(逐层显著性)
- ▶ Outlier 单独处理
- ▶ 压缩比可达最高水平
- + 极高压缩比(光滑数据)
- 计算成本较高
- GPU 支持尚不完善

变换类方法的核心: 将数据映射到频域 → 系数稀疏 → 高效编码

其他方法速览

Tucker Decomposition, Deep Learning, Bit Manipulation

HOSVD / Tucker

代表: TTHRESH

- ▶ 张量 $\rightarrow$ 核心张量 + 因子矩阵
- ▶ 跨维度长程相关性
- ▶ CR 可达 $1000\times+$
- ▶ 但迭代求解非常慢

Deep Learning

代表: AE-SZ, CoordNet

- ▶ AE/VAE
做编码器或预测器
- ▶ Online vs. Offline 训练
- ▶ 潜力大, 但训练开销高
- ▶ 误差控制尚需改进

Bit Manipulation

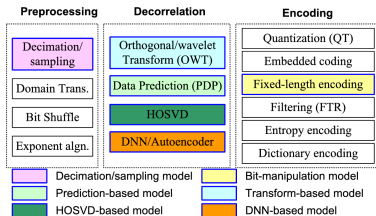
代表: SZx, cuSZp

- ▶ 截断低有效位
- ▶ 纯位操作, 极快
- ▶ cuSZp: A100 上
200-400 GB/s
- ▶ CR 相对较低

方法	速度	压缩比	误差控制
Prediction (SZ)	★★★★	★★★★★	★★★★★
Transform (ZFP)	★★★★★	★★★★	★★★★
HOSVD (TTHRESH)	★	★★★★★★	★★★★
Bit Manip (cuSZp)	★★★★★★	★★	★★★★

模块化 Pipeline 设计

How Modern Compressors Are Built



常见模块

- ▶ Predictor (PDP)
- ▶ Quantization (QT)
- ▶ Wavelet / Orthogonal Transform
- ▶ Bit-Plane Coding (BPC)
- ▶ Lossless Encoding (Huffman, Zstd)

LibPressio: 统一接口

- ▶ 一套 API 对接所有压缩器
- ▶ 自动 error bound 配置
- ▶ 语言绑定: C/C++/Python/Fortran
- ▶ Provenance 记录
- ▶ 跨压缩器性能评估

Takeaway: 现代压缩器是模块化组合, 多个技术搭配使用

九大应用领域

From Climate to Quantum Circuits

领域	驱动力	目标 CR	带宽需求	推荐压缩器
Climate	存储	$2-3 \times_L$	$> 1 \times_L$	SZ(MSZ)
Combustion	存储	$2-5 \times$	—	SZ(CliZ), ZFP
Cosmology	存储+吞吐	$> 10 \times$	$> 1 \times_L$	SZ(ROIBIN-SZ), LC
Fusion	存储+吞吐	$> 5 \times$	—	SZ(cuSZ), ZFP, LC
Light Sources	吞吐	$> 10 \times$	1 TB/s	SZ(cuSZp), ZFP
Seismology	吞吐+存储	$> 20 \times$	$> 1 \times_L$	SZ, ZFP, MGARD
Molecular Dyn.	存储	$> 3 \times$	$> 1 \times$	MGARD
System Logs	吞吐	$> 10 \times$	$> 10 \times_L$	SZ(cuSZx)
Quantum Circuit	内存	$2-10 \times$	25 GB/s	需评估

$\times_L$ = 相对最优无损压缩器的倍数

不同领域需求差异巨大: 有的要**极致 CR**, 有的要**实时带宽**

QoI: 科学家真正关心什么?

Quantities of Interest Preservation

仅保证 pointwise error bound 往往不够——科学家关心的是衍生量

三类 QoI 需求:

1. Derived quantities

dSSIM (气候), 统计量, 梯度

2. Topological features

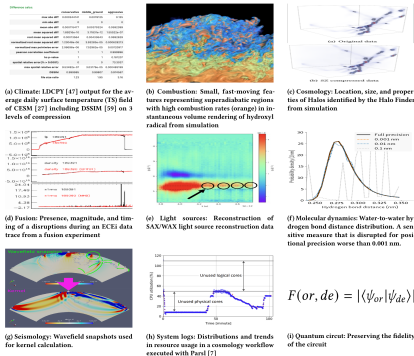
Halo (宇宙学), 峰值/扰动 (聚变)

3. Distributional features

氢键距离分布 (MD), Fidelity (量子)

9 个领域中 8 个表示现有压缩器
难以直接保持其 QoI

Gap: pointwise error $\neq$ QoI preservation — 这是重要的 open problem



$$F(or, de) = |\langle \psi_{or} | \psi_{de} \rangle|$$

拓展应用

Beyond Storage: GPU, Queries, In-situ, and More

GPU 加速压缩

- ▶ cuSZ, cuSZp: CUDA 原生实现
- ▶ cuSZp 在 A100 上达 200-400 GB/s
- ▶ 挑战: Huffman 编码在 GPU 上并行难
- ▶ 新思路: fixed-length encoding (cuSZp)

硬件可移植性

- ▶ GPU 压缩 → CPU 解压(宇宙学需求)
- ▶ FPGA 实现(光源实时需求)
- ▶ 不同平台算术一致性是难题

压缩数据上的查询

- ▶ 直接在压缩数据上查询或部分解压后查询
- ▶ 避免先全量解压再查询的开销
- ▶ 与 post analysis 深度结合

In-situ & Streaming

- ▶ 模拟运行中实时压缩 I/O 数据
- ▶ Checkpoint/Restart 加速
- ▶ 聚变等场景需要流式压缩
- ▶ 追踪数据变换历史

如何选择压缩器?

A Practical Guide

场景	推荐	原因
通用科学数据	SZ3 / QoZ	高 CR + 严格误差控制
需要 random access	ZFP (fixed-rate)	分块独立, O(1) 解压
极致压缩比	SPERR / TTHRESH	小波/Tucker, CR 最高
极致速度 (GPU)	cuSZp	200-400 GB/s
不确定选哪个	LibPressio	统一接口, 一键对比

选择三要素:

1. 数据特征: 光滑 $\rightarrow$ 变换类更好; 离散/稀疏 $\rightarrow$ prediction 更好
2. 误差需求: strict pointwise $\rightarrow$ SZ; L2 norm $\rightarrow$ MGARD/TTHRESH
3. 硬件与速度: GPU 实时 $\rightarrow$ cuSZp/cuSZ; CPU batch $\rightarrow$ SZ3/SPERR

开放问题与未来方向

What's Next?

1. QoI-native 压缩

直接优化科学量(而非仅 pointwise error)—— 大部分领域的刚需

2. 硬件异构与可移植

GPU/CPU/FPGA 跨平台压解, 算术一致性保证

3. 流式与实时压缩

1TB/s 光源数据, 在线量子模拟 —— 需要 μs 级延迟

4. Deep Learning 融合

AE/INR 方向潜力大, 但训练开销与误差控制仍是瓶颈

5. 生态与工具链

LibPressio 标准化, HDF5/NetCDF 集成

这是一个**系统 + 算法 + 应用**高度交叉的领域

友情宣传: 招生广告

以下老师的课题组正在招收博士生/硕士生/实习生, 有兴趣的同学欢迎联系!



Boyuan Zhang

张博源

Assistant Professor
U of Kentucky, CS

*Data Compression,
HPC, ML Systems,
Quantum Computing*

bozhan@iu.edu



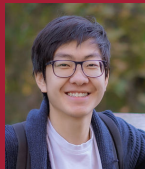
Yafan Huang

黄亚凡

Incoming Asst. Professor
UMD, CS (Fall 2026)

*AI+HPC Co-design,
Data Compression,
Fault Tolerance*

yafan-huang@uiowa.edu



Sian Jin

靳思安

Assistant Professor
Temple University, CIS

*Scientific Data Analytics,
Lossy Compression,
Parallel I/O Systems*

sian.jin@temple.edu