Data compression for HEP data

Antonio Boveia

Ohio State University / ATLAS

Input from Genesis Phase-I proposal collaborators, especially C. Doglioni, T. Elliott, A. Gupta, S. Sengupta (Manchester); M. Montella (Ohio State).
Phase-I proposal team: OSU + Manchester + BNL + Argonne + Fermilab + LANL + Indiana (ATLAS, CMS, DUNE, LHCb)

ML compression for real-time analysis and HL-LHC data volumes

HL-LHC projections: flat computing budgets vs. growing datasets.

Multiple experiments are pursuing real-time analysis strategies that trade event size for rate, broadening discovery potential
Individually small events but large volume (PBs/year).

ATLAS (TLA)	CMS (Scouting)	LHCb (Real-time)	DUNE (streaming)
Record EF trigger-level objects at 400 kHz (Goal: ~50% of events seen by EF)	L1 scouting + parking streams (100's of kHz of lightweight events)	30 MHz real-time reconstruction via Allen/GPU	Streaming readout, 100+ interactions per spill, full ND-LAr waveforms (~1 PB/yr)

Better lossless compression (e.g. integrated in ROOT, or for experimental raw data) would

- allow **more extensive science with (near) real-time analysis streams** at fixed bandwidth (searches for BSM, calibration datasets, etc.)
- **reduce data storage costs (and related overhead)** across experiments

Even modest compression gains multiply across all of these programs in many ways
(for instance, Monday's discussion about hosting open data).

Three approaches to reducing data volume

(a) Don't store what you don't need

Eliminate needless precision (mantissa truncation), drop unused variables, skim, etc.
Similar issue from Monday's session: tokenization losses should stay below experimental uncertainties (Kagan)

(b) Better classical algorithms and data formats

Columnar data formats and modern compressors (e.g. ZSTD) are showing promise over legacy baselines (TTree + LZMA).
Active R&D across experiments.

(c) ML-based compression (lossy and lossless)

Can machine learning exploit the statistical structure of HEP data to compress beyond what traditional methods achieve?

Connected directly with tokenization / foundation model approaches, though with different goals.
How do compression models pre-trained broadly across experiments perform compared to per-application training?
Are the representations these learn useful for downstream analysis tasks?

Can ML go beyond a+b?

Lossy compression: autoencoders

(Familiar story to people here)

Simple example A 4-vector (E, p_x, p_y, p_z) through an autoencoder with a 3-node bottleneck.

- Train on massless particles. The network learns $E^2 = p^2c^2$.
- Nonlinear generalization of PCA. Keep the dominant components, discard less important.
- Feed it a massive particle → reconstruction breaks, or reduce the latent space further and the resolution on (p_T, η, ϕ) degrades beyond tolerable reconstruction error.

Inference (where compression/decompression happens) can be made relatively fast

- Achieve compression via ML overfitting. Train on all data first, then the neural network is used for compression of the same data. First presented in ATLAS in 2019!
- **Dangerous for compression** when the input is anomalous w.r.t. training data.
- **Preliminary results available for online compression (BALER).** Requires an embedded anomaly detection step **as data comes in, remove outliers.** Results OK but not as good as offline compression (**CHEP2023**).

Lossy methods fine for many applications, but not for raw data streams.

- Stress tests using ATLAS open data by Manchester et al. in 2024 (only photon kinematics).
- Can we rediscover the Higgs from compressed data? BALER shifts the Higgs mass from 125.0 → 123.9 GeV. Physics distortion too large, even before throughput optimization.

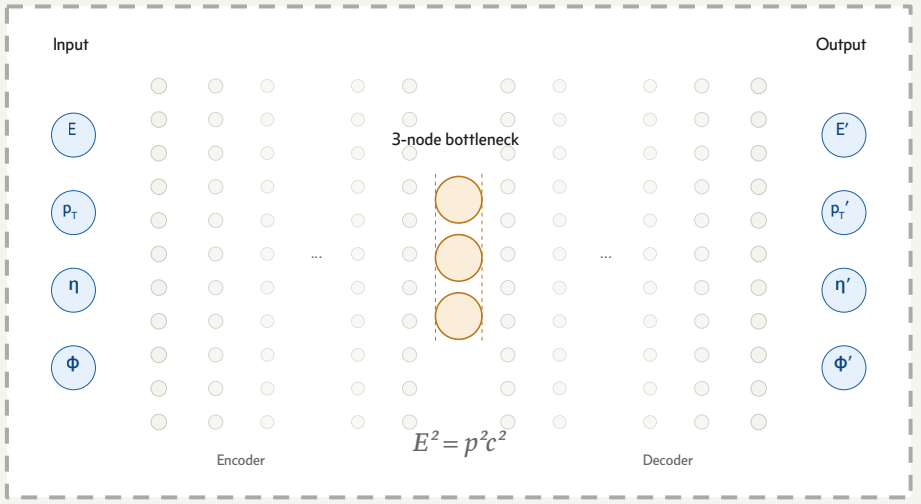


Table 3: Performance Metrics

Method	Compressed Size (MB)	Compression Ratio	Compression Throughput (MB/s)	Decompression Throughput (MB/s)
Baler AE ratio 1.6:1	47.276	1.30:1	1.028	1.007
Baler AE ratio 5:1	17.73	3.47:1	2.750	3.199
Blosc2 Prec. 12	37.71	1.63:1	106.329	384.558
Blosc2 Prec. 14	42.14	1.46:1	118.936	466.131
SZ3 Abs. 1e-2	30.93	1.99:1	9.226	8.595
SZ3 Rel. 1e-5	13.22	4.65:1	26.295	11.508
zfp Prec. 18	33.71	1.82:1	66.303	64.093
zfp Tol. 1e-3	46.15	1.33:1	90.307	70.561

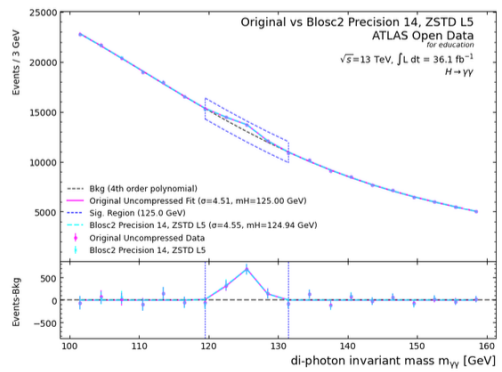


Figure 10: Experiment results - Best Overall: Blosc2, Precision 14 vs baseline.

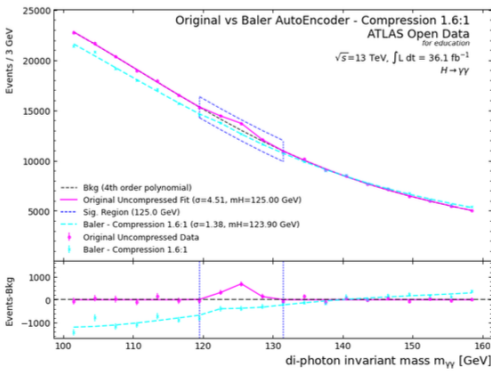


Figure 11: Experiment results - Baler Compression Level 1.6:1 vs baseline.

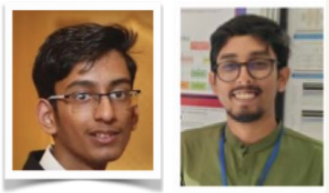
Bradley Booth

Autoencoders for compression A. Boveia · Snowmass Computational Frontier ML Group, November 2020
The Dark Machines anomaly score challenge A. Aarrestad et al. · SciPost Phys. 12 (2022) 043
Student theses including
Eric Wallin (2020) — "Tests of Autoencoder Compression of Trigger Jets in the ATLAS Experiment"
Eric Wulff (2020) — "Deep Autoencoders for Compression in High Energy Physics"
Honey Gupta (2020, Google Summer of Code, IIT Madras) — Autoencoder compression for HEP data.
Love Kildetoft (2021) — "Evaluation of float-truncation based compression techniques for the ATLAS jet trigger"
Jessica Lastow (2021) — "Investigation of Autoencoders for Jet Images in Particle Physics"
Other literature exists on AE/neural-network-based forms of compression in HEP and beyond;
see e.g. 2210.11489, 2411.01118v2, 2402.05013, 2602.15751, 2105.01683

Blosc2 (acceptable) vs. Baler AE (m_H shifts 125→124 GeV)]
Bradley Booth, DeepMind Research Internship 2024

Lossless compression: the BOA Constrictor

Gupta, Doglioni, Elliot, Sengupta
arXiv:2511.11337, accepted by MLST



Akshat Gupta
Sanjiban Sengupta

Mamba-based models represent a powerful new approach to lossless compression, exploiting long-range detector and physics correlations that classical algorithms cannot

- Classical compressors typically use fixed, short-context dictionary methods
- These are blind to the long-range correlations in physics data (patterns in raw readout, conservation laws, detector geometry, kinematic structure)
- Can autoregressive ML models learn these correlations, and approach the Shannon entropy floor more efficiently?
- Does this theoretical advantage translate to gains at acceptable throughput?

In BOA, ML is cleanly separated from the encoding

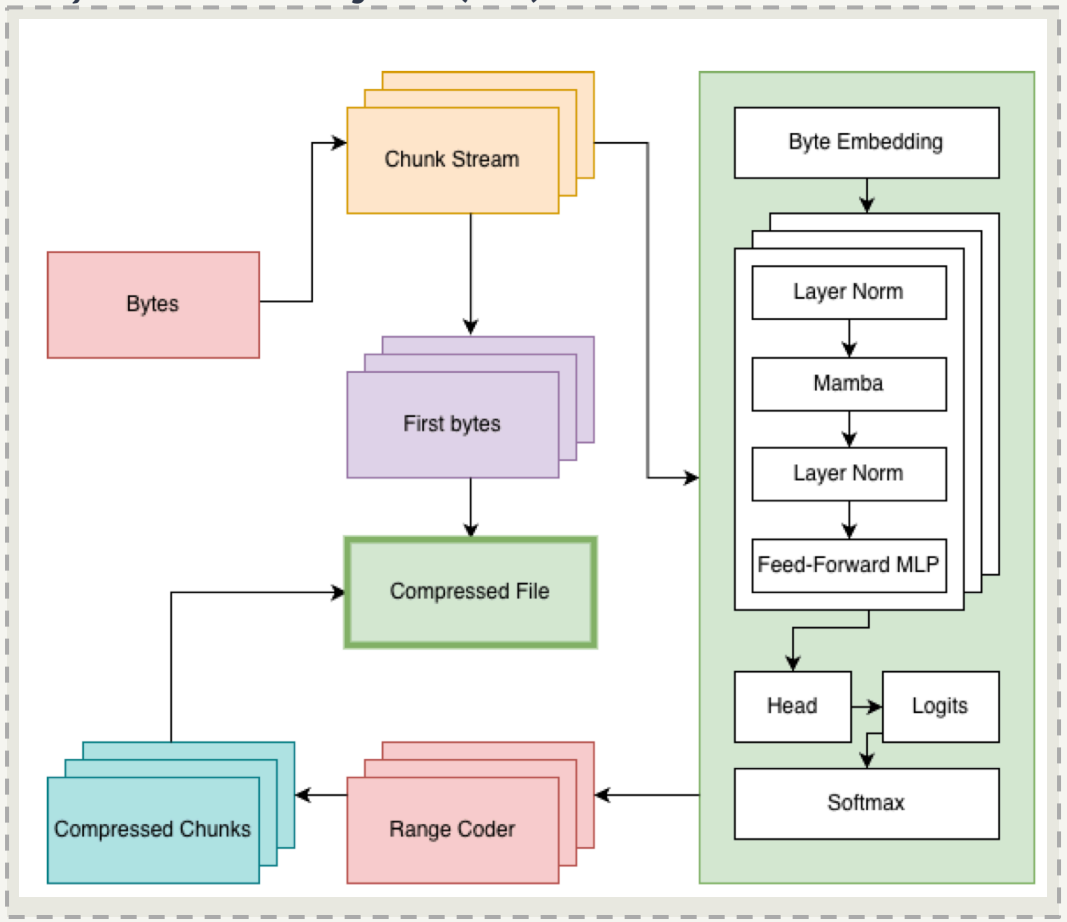
- The ML (Mamba SSM, faster training and inference than transformers) learns how to predict next bit/byte/token.
- A **range coder** (1970s tech) does the actual entropy coding using $-\log_2 P(\text{byte} \mid \text{context})$ bits. **The compression is guaranteed bit-exact.**
- If the model is confident and correct (high probability of the right byte), this takes very few bits. If the model was uncertain or wrong, more bits needed. *Let's play 20 questions…*

The ML never touches the data representation

Practical considerations

- Training on dataset in advance or on the fly (complication for decoding; same weights needed for compression & decompression). *Streamline and online prototypes* available.
- Price to pay: (de)compression speed [working on this within NextGen WG 1.7 / ROOT]

Bytewise Online Autoregressive (BOA) constrictor



BOA results across datasets

HEPMC

Algo.	Size (MiB)	Ratio	C (MiB/s)[1]	D (MiB/s)[1]
Baseline	1992.24	N/A	N/A	N/A
LZMA(9)	373.64	5.33	0.95	92.49
ZLIB(9)	497.83	4.00	12.76	369.61
ZSTD(1)	485.88	4.10	**633.93**	1289.38
ZSTD(3)	519.64	3.83	305.53	994.42
ZSTD(7)	477.31	4.17	101.22	981.68
ZSTD(19)	392.79	5.07	1.42	1003.01
LZ4	862.77	2.31	589.32	1271.19
LZ4 (HC)	614.95	3.24	15.54	**1355.04**
Brotli(1)	532.54	3.74	301.18	389.35
Brotli(11)	374.10	5.33	0.72	366.03
bzip2(9)	387.52	5.14	19.69	55.38
BOA[2]	**252.88/253.99**	**7.87/7.84**	12.63	8.21
BOA[3]	**221.86/224.07**	**8.98/8.89**	9.54	6.34
BOA[4]	**216.02/218.25**	**9.22/9.13**	9.32	6.90

[1] Compression & Decompression Throughput
[2] 128x2; Trained for 20ep with 20% data
[3] 256x1; Trained for 10ep with 200MiB data
[4] 256x1; Trained for 30ep with 200MiB data

NanoAOD

Using raw buckets for extraction from ROOT format to standalone

Dataset	BOA	LZMA-9	ZLIB-9
NanoAOD dataset			
JetHT (2016G, orig.)	**5.58**×	5.40×	3.21×
JetHT (2016G)	**5.63**×	5.47×	3.23×
CharmoniumTrigger	**7.50**×	7.46×	4.22×
ElectronTrigger	**6.59**×	6.50×	3.53×
SingleMuonTrigger	6.87×	**6.92**×	3.80×
JetHT (2016H)	**5.70**×	5.55×	3.29×
MiniAOD			
JetHT (2016G)	1.69×	**7.11**×	5.77×
JetHT (2016H)	1.82×	**8.68**×	6.76×

Not expecting BOA to do well from Nano → MiniAOD, as it learns data structure patterns

ATLAS ttbar (HDF5)

Algo.	Size (MiB)	Ratio	C (MiB/s)[1]	D (MiB/s)[1]
ATLAS (Row-Major)				
Baseline	546.63	N/A	N/A	N/A
LZMA(9)	322.42	1.70	1.59	36.07
ZLIB(9)	338.86	1.61	19.73	346.09
ZSTD(1)	340.33	1.61	649.99	1343.24
ZSTD(3)	340.28	1.61	318.92	1273.32
ZSTD(7)	333.86	1.64	114.21	1429.57
ZSTD(19)	334.52	1.63	3.69	859.29
LZ4	384.41	1.42	**868.02**	1763.46
LZ4 (HC)	365.42	1.50	13.70	**1817.65**
Brotli(1)	347.98	1.57	290.35	239.27
Brotli(11)	322.52	1.69	0.56	170.56
bzip2(9)	337.84	1.62	21.71	31.43
BOA[2]	**251.40/252.51**	**2.17/2.17**	11.83	7.87
BOA[3]	**247.29/249.49**	**2.21/2.19**	9.14	6.42
ATLAS (Col-Major)				
Baseline	546.63	N/A	N/A	N/A
LZMA(9)	**285.30**	**1.92**	1.81	40.24

All results from CMS and ATLAS Open Data, HEPMC public generator output. Source: arXiv:2511.11337 (v2 forthcoming)

- All testing so far is standalone (extract from ROOT format into "bytes" in different ways, then compress)
- BOA matches or outperforms LZMA-9.
- Gains are largest on highly structured formats (HEPMC), more modest on complex detector data (ATLAS $t\bar{t}$).
- Training generalizes across different trigger streams on CMS Open Data. No per-dataset retraining required.

- Throughput not yet competitive with highly optimized compressors (BOA's GPU/C++ classes not optimized at all yet). This will be among the deciding factors on whether worth pursuing in production.

Compression / tokenization working points

Compression ratio measures how well the model has learned the statistical structure of the data (better prediction → higher compression ratio).
It tells you how much predictable structure the model found (detector geometry, kinematic correlations, detector readout patterns, invariance, etc.)
High compression = the model learned to predict well.
Low compression = near entropy floor given the model's capacity.

Different data, different working points

BOA generalizes well across CMS trigger streams (trained once, compresses JetHT, Charmonium, Electron, Muon) but not always broadly (miniAOD).
Compression ratios vary with data structure (modest for complex detector data, dramatic for highly structured formats).

Raw detector readout (zero degradation required) is the hardest and highest-payoff target for lossless methods.
Derived formats and open data benefit from lossless compression and can additionally tolerate controlled-precision loss.
Hardware-constrained edge compression (on-detector, real-time) operates in a different regime entirely: extreme lossy compression under low latency.
These are different operating points on the same spectrum.

Connection to tokenization

- Both compression and foundation models learn the same underlying statistical structure but for different purposes (storage vs. task performance). Whether the optimal representations for storage and for task performance converge/diverge is an open question.
- We want to understand how close ML methods approach the fundamental limits giving us both a theoretical ceiling and more insight into the structure of our data.
- Working with information theory experts at OSU and Manchester.

Planned work

Systematic benchmarking

BOA and 13+ classical algorithms (ZSTD, LZ4, LZMA, Brotli, SZ3, zfp, and others) on real data from ATLAS, CMS, LHCb, and DUNE from raw data to ntuples.
Mapping where ML exceeds the classical ceiling and where it doesn't, across data formats and experiments.
The model weights (~2 MB) ship with the compressed data. BOA compresses in chunks; random access to events does not require decompressing the full file.

Representation and architecture optimization

Currently BOA operates on ad-hoc byte-level serialization of ROOT data. This is almost certainly suboptimal.
How does data representation affect compressibility?
Are tokenized representations (like TREASURE's) close to incompressible or do they compress further?
Explore alternatives to Mamba (e.g. RNNs), scaling-law analysis of how performance grows with model size and training data.

Throughput optimization

This is the main challenge. For now, BOA's unoptimized throughput is orders of magnitude below ZSTD.
What is the trade-off between CPU/GPU/latency and disk?
Working on inference acceleration with the ROOT/NextGen team, quantization, architecture simplification.
In which applications can this be made production-viable?

Characterizing the fundamental limits

Working with an information-theory collaborators to understand how close these methods are to the best compression achievable on these datasets.

We have applied for Phase I Genesis funding to support this work.

Summary

Both compression and tokenization try to answer *how do you represent this data so a model can predict it efficiently?*
Tokenization is lossy compression optimized for reconstruction fidelity; BOA is lossless compression optimized for storage.

ML compression techniques offer the potential for more efficient bit-exact data storage, which would have a broad impact on the physics capabilities of HEP experiments in general and the HL-LHC in particular.

There is a large intersection between our compression program and TREASURE's tokenization R&D.

Some examples from this week's discussion:

- Joint optimization of tokenization for task performance and compressibility
- Cross-experiment transfer and compression as a foundation model
- Compression as practical infrastructure for open data hosting and distribution
- etc.

We're keen to collaborate!