



Evaluating Lossy Compression Effects on Generated Physics Objects in RNTuples

August 10th, 2026

Nicholas Barnett (University of Illinois Chicago), Austin Baty (University of Illinois Chicago),
Philippe Canal (Fermilab), Florine Willemijn de Geus (CERN)

Project Inspiration

- Data collected from particle colliders increasing in size
 - Use compression algorithms to store and collect more data
 - Lossless compression already widely used, especially in ROOT
- Upgrades to colliders and experiments require improved information storage
 - Hi-Lumi LHC will produce more data than LHC has
 - EIC and FCC will need to store huge information amount when operational
- Lossy compression allows much more information storage
 - Investigating and understanding effects of lossy compression on physics observables is needed

Lossy Compression Algorithms

- **Truncation** - Drops bits off mantissa for value, only lowers values
- **Quantization** - Maps value to bin index in range, fixed bin-width error
- **SZ3** - Five stage compression process with user provided error bound

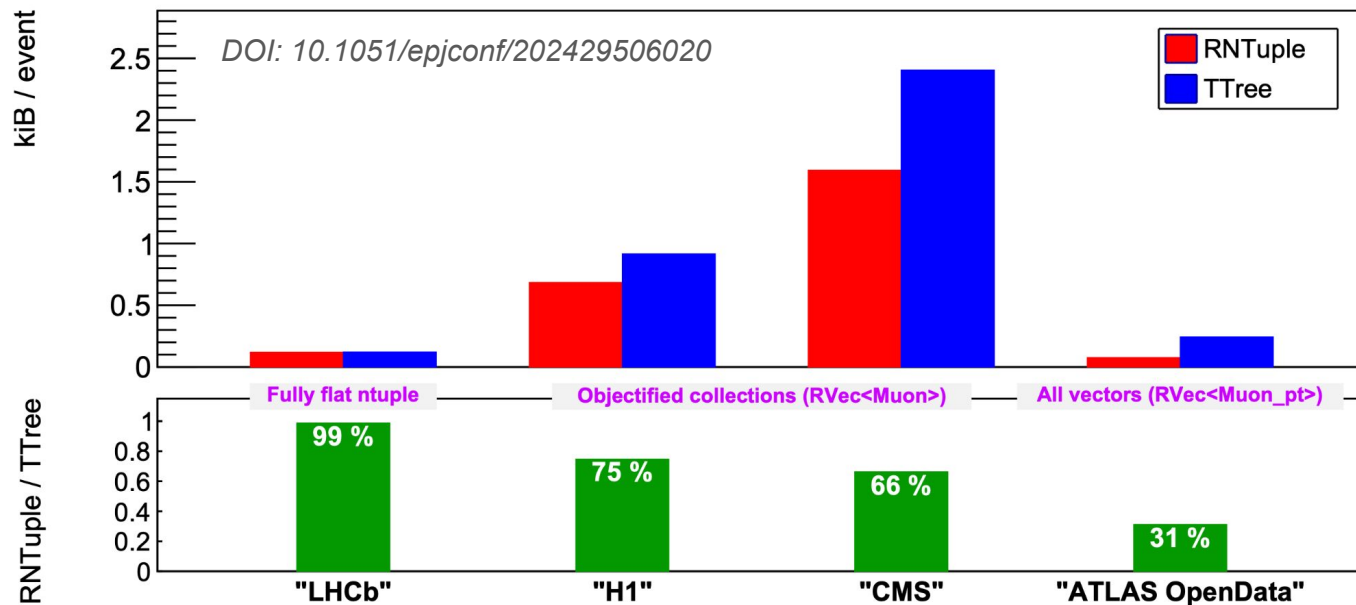
Physics Objects



- Pythia generated samples of four collision systems
 - LHC pp Hard QCD, LHC pp $W \rightarrow e + \nu$, EIC ep neutral-current DIS, FCC ee Z-pole
- Physics objects made from generated particles for each event
 - Jets (with FastJet), thrust, missing transverse energy (MET), sphericity
 - Using generator level objects to be experiment agnostic
- Observable metrics from objects include response and angular resolution
 - Response: $p_{T,\text{compressed}}/p_{T,\text{uncompressed}}$
 - Angular resolution: $\Delta R(\text{compressed, uncompressed})^2 = \Delta\eta^2 + \Delta\phi^2$
 - Compression ratio: compressed file size/ uncompressed file size

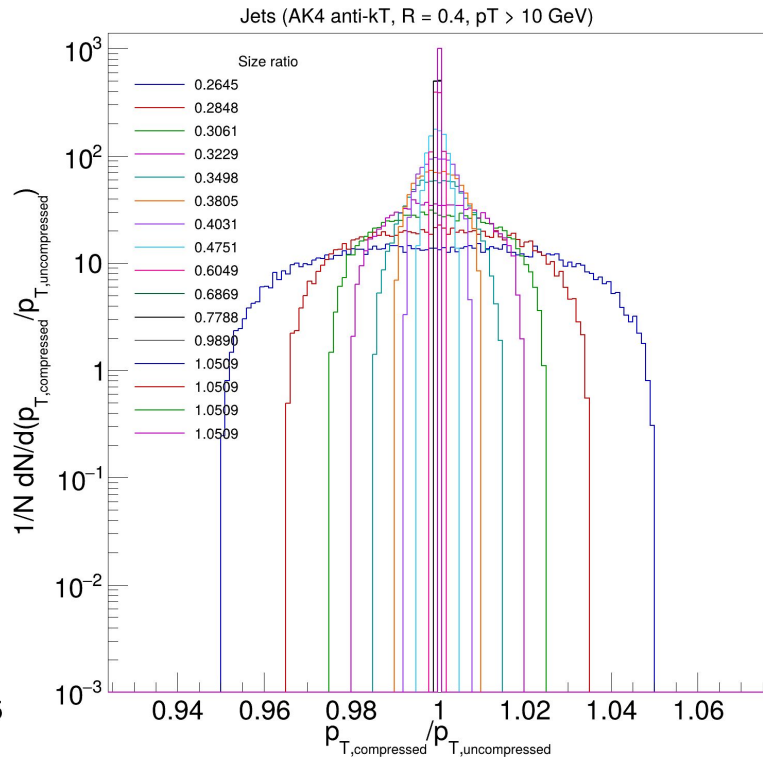
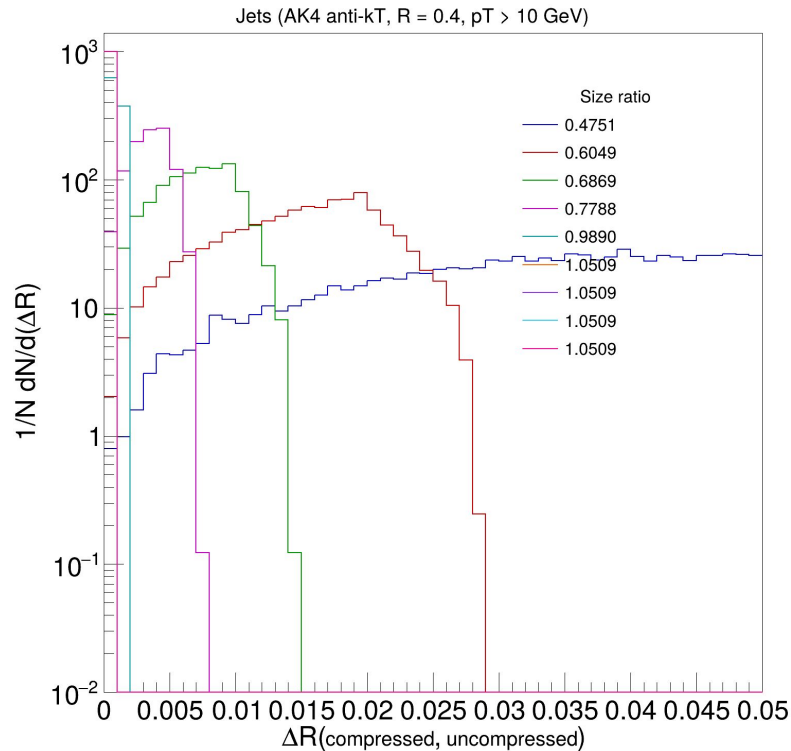


RNTuple



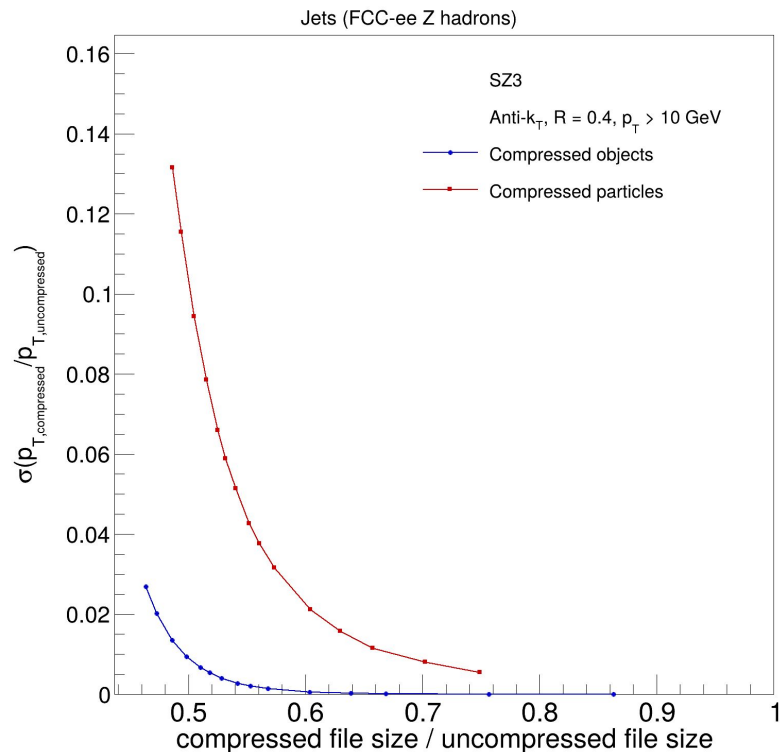
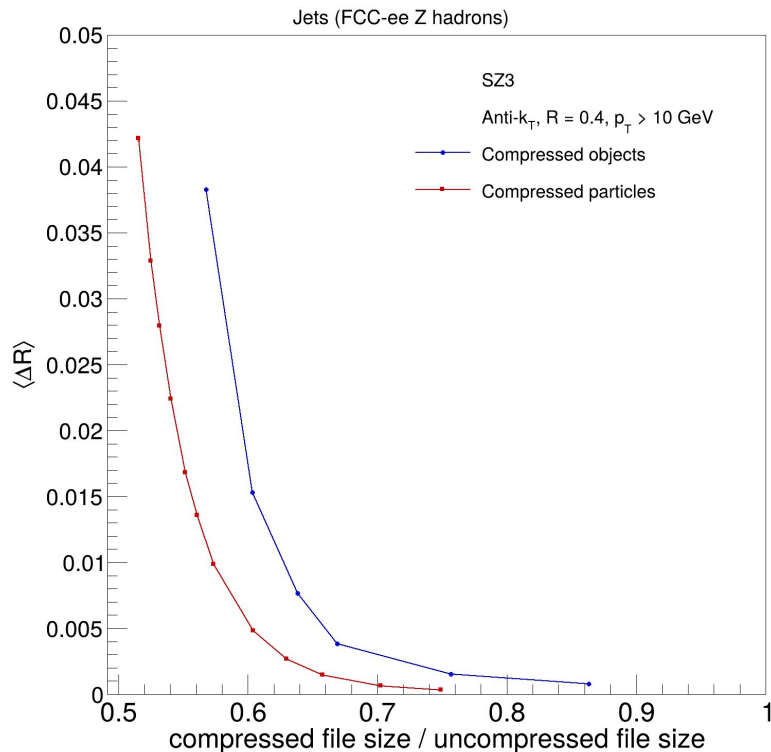
- RNTuples contain same information as TTrees, with less storage space
 - Reading and writing RNTuples is faster than TTrees
- RNTuple is intended to supersede TTree in ROOT

Compression Object Distributions



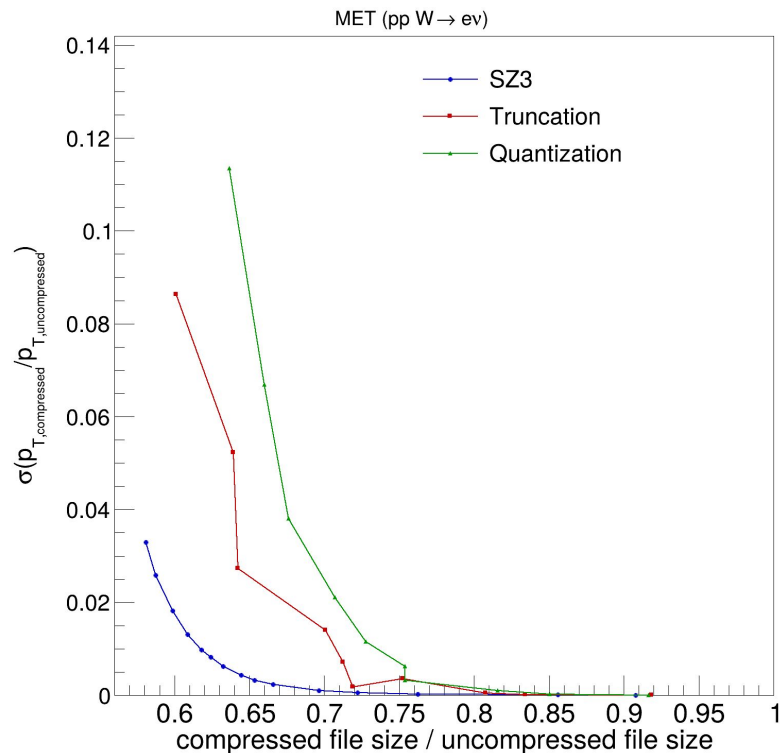
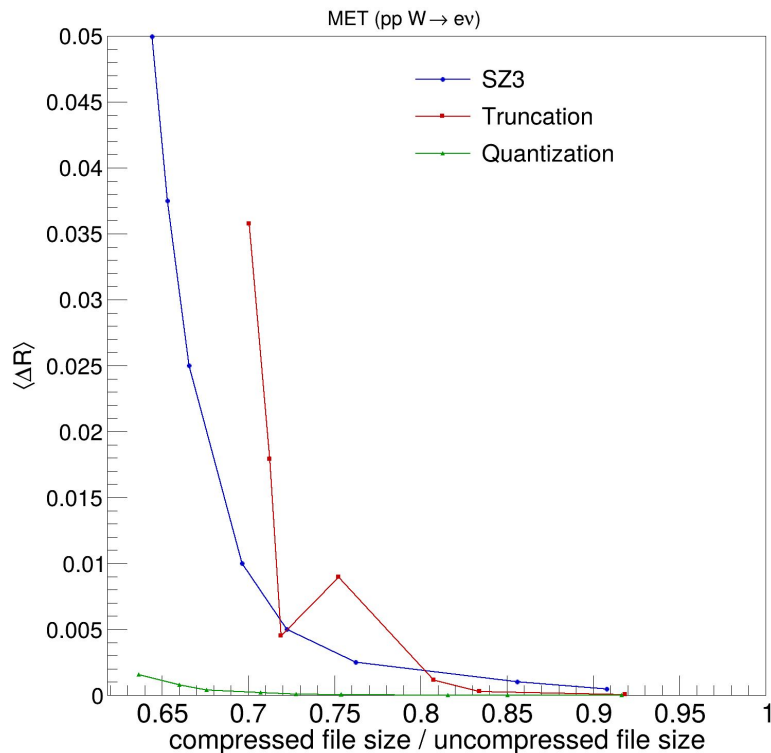
- Observing these distributions for jets, thrust, missing transverse energy, and sphericity
 - Showing SZ3 compressed jets in LHC like sample

Comparing Compression by Implementation



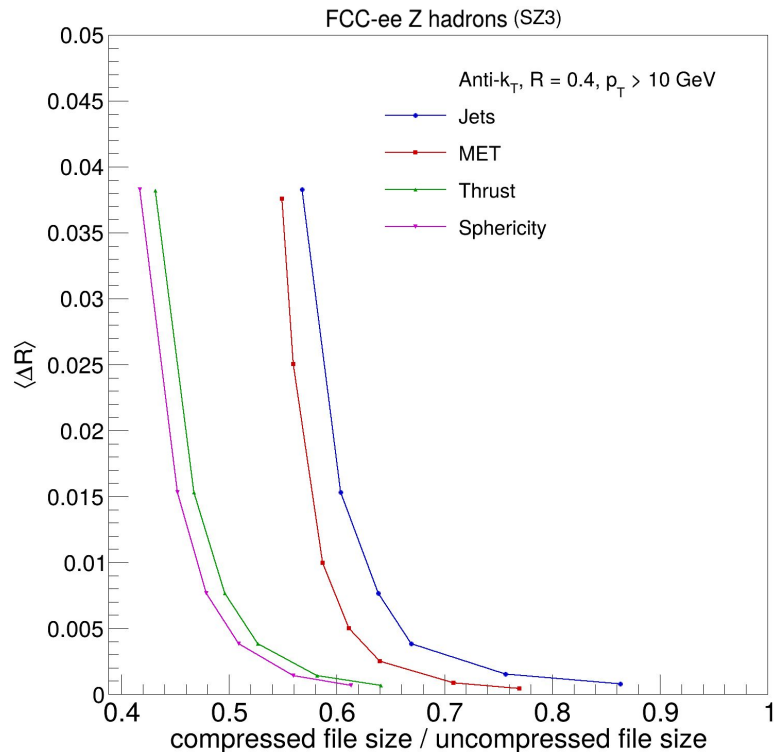
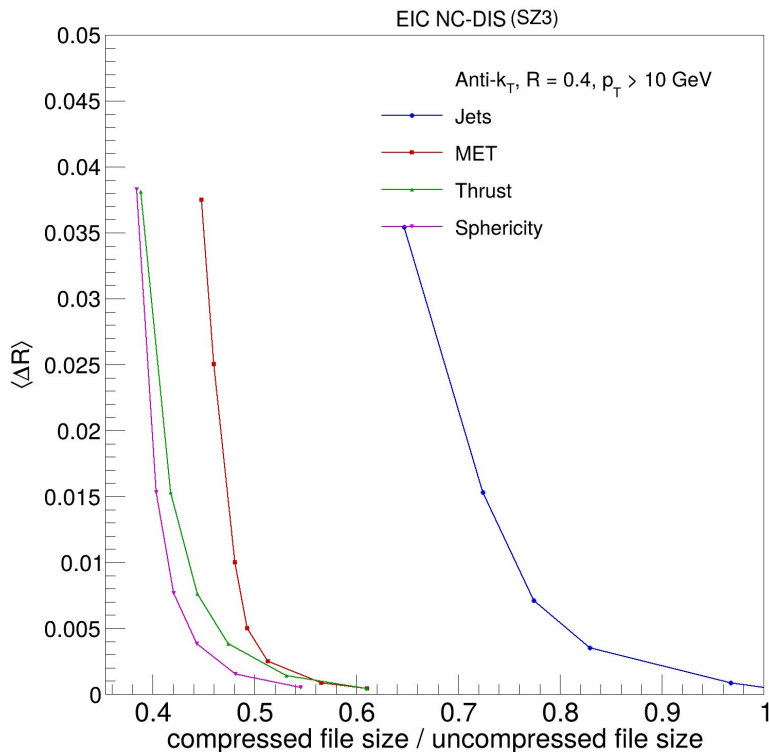
- Clustering jets from compressed particles smears p_T less than compressing clustered jets, but gives slightly worse angular resolution

Comparing Compression by Algorithm



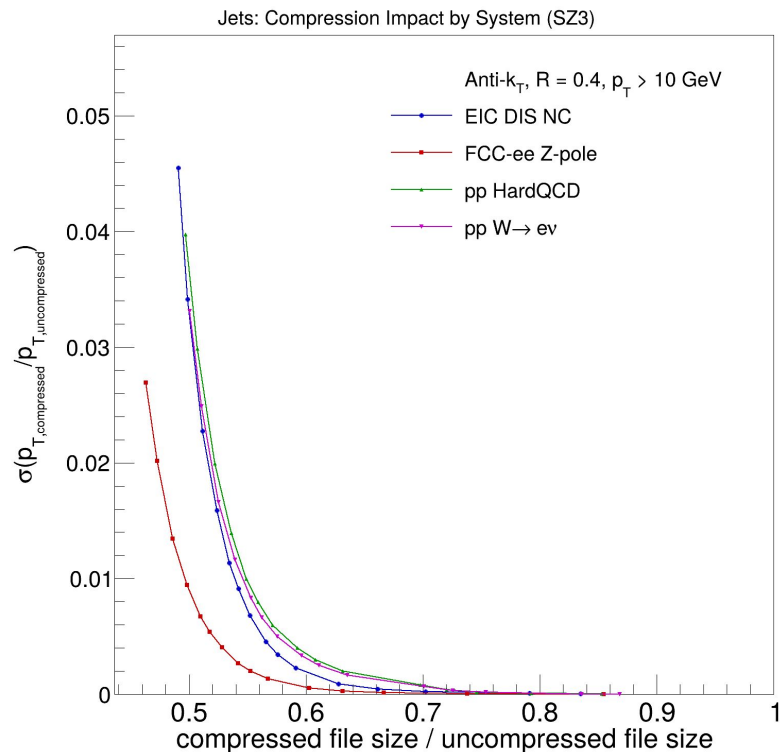
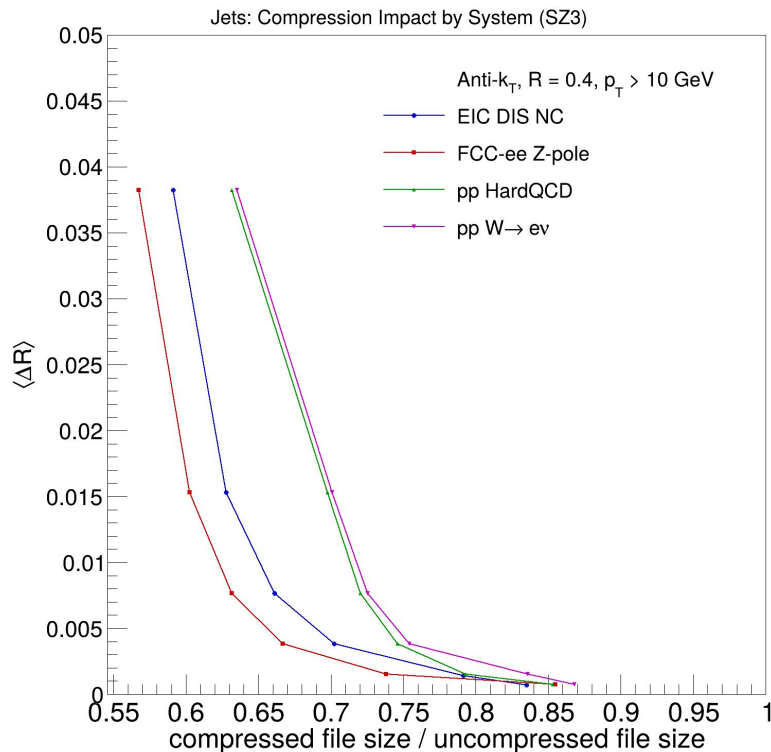
- Quantization degrades abject axis least, but effects p_T resolution most
 - SZ3 seems to affect objects less than truncation

Comparing Compression by Object



- Can see differences in compressed objects, for the same system, using different algorithms
 - Jets seem to be more affected by lossy compression than other objects

Comparing Compression by System



- Effect of compression similar across different samples generally
 - FCC like sample shows slightly more robustness against compression effects

Summary

- Determined and showed metrics to quantify effect of lossy compression algorithms on physics observables
 - Angular resolution and response show effects of compression on objects
 - Reproducible and similar results across different event samples
- Looking for best implementation of lossy compression on physics data
 - Compressing particles or objects
 - How each object performs before and after compression
 - Object and collision system dependence on different algorithms
- Preliminary results show jets are more impacted than other objects, and compressing objects with SZ3 on objects tends to give better results
- Presented this work at New Perspectives conference this year, which took place at Fermilab on July 16th

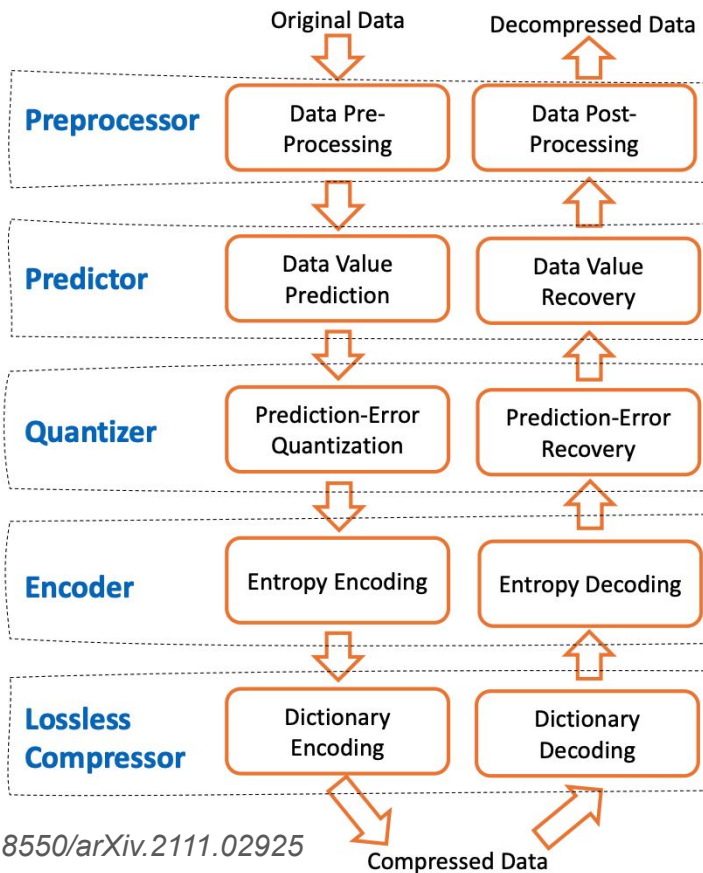
Next Steps

- Determine best visualization of these plots
 - Overlays shown by system, object, compression algorithm, and implementation
- Investigate population loss of objects when compressing particles before building objects
 - See greater loss of object count after quantization than other algorithms
- Writing paper currently to convey what's seen in these distributions
 - Where, how, and which lossy compression algorithm is best implemented
 - Quantify fidelity of objects after the compression

Backup

SZ3

Pipeline/Module Abstraction



- Preprocessor: transform data before prediction
- Predictor: estimate values based on local data
- Quantizer: map prediction error to bins (introducing error), enforce error bound
- Encoder: lossless entropy coding of integers into binary
- Lossless Compressor: additional lossless compression on output

Pythia Samples

LHC pp Hard QCD

- Beam ids = 2212
 - proton-proton
- $E_{\text{CM}} = 13.6 \text{ TeV}$
- pTHatMin = 15
- HardQCD:all = on
($q\bar{q} \rightarrow q\bar{q}$, $qg \rightarrow qg$, $gg \rightarrow gg$, $gg \rightarrow q\bar{q}$, etc.)

LHC pp $W \rightarrow e + \nu$

- Beam ids = 2212
 - proton-proton
- $E_{\text{CM}} = 13.6 \text{ TeV}$
- ffbar2W
 - W decay, only channels with an e or e, ν

EIC neutral-current deep inelastic scattering (NC-DIS)

- Beam idA = 2212 (proton)
 - $E_A = 275 \text{ GeV}$
- Beam idB = 11 (electron)
 - $E_B = 18 \text{ GeV}$
- ff2ff(t:gmZ)
 - t-channel photon/Z exchange
- Q2Min = 25 GeV^2

FCC ee Z-pole hadrons

- Beam ids = -11,11
 - e^+e^-
- $E_{\text{CM}} = 91.19 \text{ GeV}$ (Z mass)
- ffbar2gmZ
 - Only hadronic Z decays
 - s-channel fermion-antifermion annihilation

Pythia Samples

LHC pp Hard QCD

Beams:idA = 2212
Beams:idB = 2212
Beams:eCM = 13600.0
HardQCD:all = on
PhaseSpace:pTHatMin = 15.0

LHC pp $W \rightarrow e + \nu$

Beams:idA = 2212
Beams:idB = 2212
Beams:eCM = 13600.0
WeakSingleBoson:ffbar2W = on
24:onMode = off
24:onlyAny = 11 12

EIC NC-DIS

Beams:frameType = 2
Beams:idA = 2212, Beams:eA = 275.0
Beams:idB = 11, Beams:eB = 18.0
WeakBosonExchange:ff2ff(t:gmZ) = on
PhaseSpace:Q2Min = 25.0
SpaceShower:dipoleRecoil = on
SpaceShower:pTmaxMatch = 2
PDF:lepton = off
TimeShower:QEDshowerByL = off

FCC ee Z-pole hadrons

Beams:idA = -11
Beams:idB = 11
Beams:eCM = 91.1876
PDF:lepton = off
WeakSingleBoson:ffbar2gmZ = on
23:onMode = off
23:onlyAny = 1 2 3 4 5