Lossy Compression Artifacts and Approaches to "Decompress"

August 27, 2025

Yolanne Lee
University College London

Introduction

Motivation and Context

- Lossy compression in ATLAS
- What is lost?
- Can neural methods accommodate as a post-hoc measure to recover lost data?

Introduction

Context

Lossy Compression

Currently implemented lossy compression reduces the precision of a 32-bit IEEE 754 floating-point number by **truncating** its mantissa to a specified number of bits while applying **rounding** to the nearest representable value.

A 32-bit IEEE 754 float consists of:

$$\text{bits} = \underbrace{s}_{\text{1 bit}}\ \underbrace{e}_{\text{8 bits}}\ \underbrace{m}_{\text{23 bits}}$$

where:

- s is the sign bit,
- e is the exponent in *biased form* ($e \in [0, 255]$),
- m is the mantissa (fraction) with an implicit leading 1 for normalized values.

Lossy Compression

- *If x is NaN or ∞, return x.*
- *Decompose x into*

$$x = (-1)^s \cdot (1.m) \cdot 2^{(e-127)},$$

  where s is the sign, e is the biased exponent, and m is the 23-bit mantissa.
- *Let $k = 23 - n$ denote the number of mantissa bits to remove.*
- *Round m to n bits by adding 2^{k-1} to m (nearest-neighbor rounding).*
- *Zero out the lowest k bits of m (truncate).*
- *Reconstruct*

$$\tilde{x} = (-1)^s \cdot (1.\tilde{m}) \cdot 2^{(e-127)},$$

  where $\tilde{m}$ is the rounded n-bit mantissa.
- *Return $\tilde{x}$.*

Introduction

Data Exploration

Compression Performance

Type	Avg Original Size (MB)	Avg Compressed Size (MB)	Compression %
Real Pairs	225.27	167.10	25.82%
Sim Pairs	742.51	539.38	27.36%

Real Data (Averaged over 2 files)

	% Branches		% Size		Count		Size (MB)	
	Comp	Orig	Comp	Orig	Comp	Orig	Comp	Orig
Data Type								
float32 (>f4)	34.92	34.92	57.96	67.88	276.50	276.50	91.53	147.30
object_container	21.00	21.13	15.96	12.74	218.50	219.50	30.72	32.95
jagged_array	13.72	13.72	8.50	6.41	108.50	108.50	13.32	13.66
strided_object	12.93	12.93	6.22	4.69	133.00	133.00	13.37	13.44
uint32 (>u4)	10.66	10.54	7.68	5.52	84.50	83.50	12.27	11.90
int32 (>i4)	2.22	2.22	2.94	2.20	17.50	17.50	4.73	4.81
uint64 (>u8)	0.76	0.76	0.57	0.42	6.00	6.00	0.90	0.90
AsDtype	3.15	3.15	0.15	0.12	25.00	25.00	0.22	0.24
int64 (>i8)	0.12	0.12	0.02	0.02	1.00	1.00	0.04	0.06
group	0.12	0.12	0.00	0.00	1.00	1.00	0.00	0.00
unreadable_branch	0.38	0.38	0.00	0.00	3.00	3.00	0.00	0.00

Simulated Data (Averaged over 2 files)

	% Branches		% Size		Count		Size (MB)	
	Comp	Orig	Comp	Orig	Comp	Orig	Comp	Orig
Data Type								
float32 (>f4)	32.52	32.52	52.28	64.22	416.00	416.00	287.60	485.23
object_container	19.36	19.43	17.80	14.35	295.50	296.50	92.38	100.74
uint32 (>u4)	18.09	18.01	13.37	9.10	231.50	230.50	71.52	66.92
int32 (>i4)	7.47	7.47	5.93	4.38	95.50	95.50	32.80	33.27
jagged_array	8.97	8.97	5.02	3.78	114.50	114.50	27.48	28.50
strided_object	10.62	10.62	4.11	3.08	161.50	161.50	19.50	19.58
float64 (>f8)	0.94	0.94	0.78	0.58	12.00	12.00	4.26	4.31
uint64 (>u8)	1.02	1.02	0.68	0.50	13.00	13.00	3.69	3.74
AsDtype	0.47	0.47	0.02	0.02	6.00	6.00	0.10	0.12
int64 (>i8)	0.08	0.08	0.01	0.01	1.00	1.00	0.07	0.11
group	0.08	0.08	0.00	0.00	1.00	1.00	0.00	0.00
unreadable_branch	0.39	0.39	0.00	0.00	5.00	5.00	0.00	0.00

Analysis

Analysis

Quantisation Artifacts

Quantisation Artifacts in Log Scale

- Compared original vs DL10-compressed PHYSLITE files.
- Focus on electron p_t, η, ϕ (jets and muons show similar patterns).
- Absolute differences $|x_{\text{orig}} - x_{\text{comp}}|$:
- Log-binned histograms show peaks spaced evenly in $\log_{10}$.
- Peak spacing: $\Delta \log_{10} \approx 0.28 \Rightarrow$ ratio ≈ 1.9.

Quantisation Artifacts in Log Scale

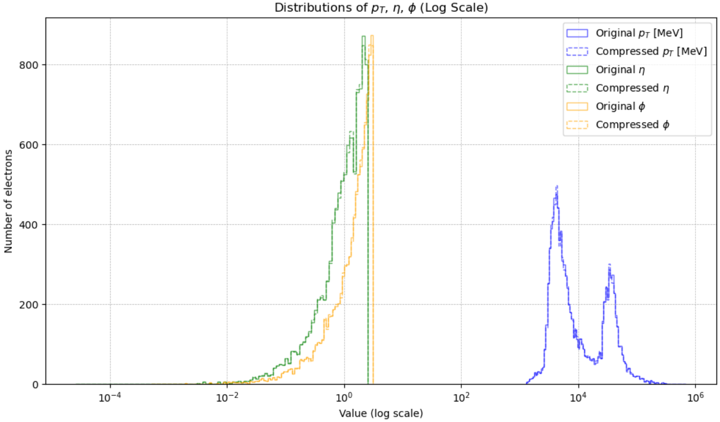


Figure 1. Distributions of values of interest.

Quantisation Artifacts in Log Scale

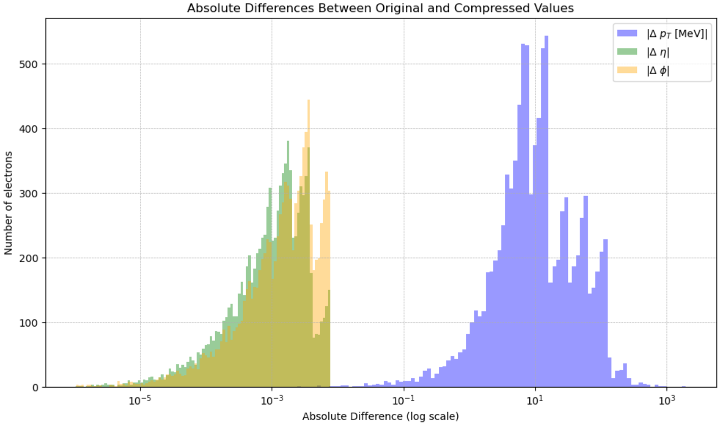


Figure 2. Log-binned absolute residuals with clear quantisation steps.

Quantisation Artifacts in Log Scale

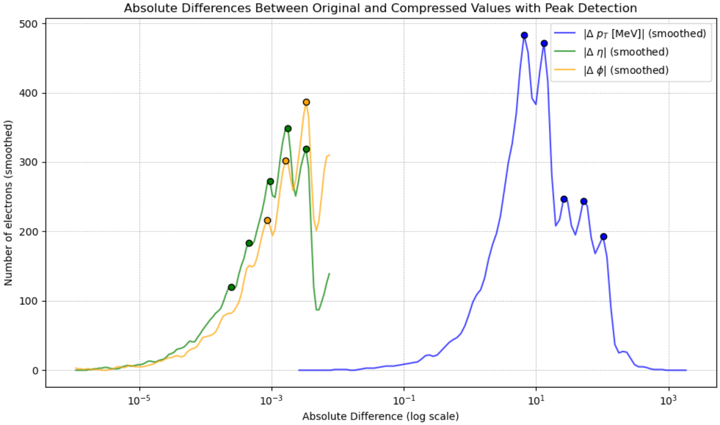


Figure 3. Log-binned absolute residuals smoothed with peak-finding.

Quantisation Artifacts in Log Scale

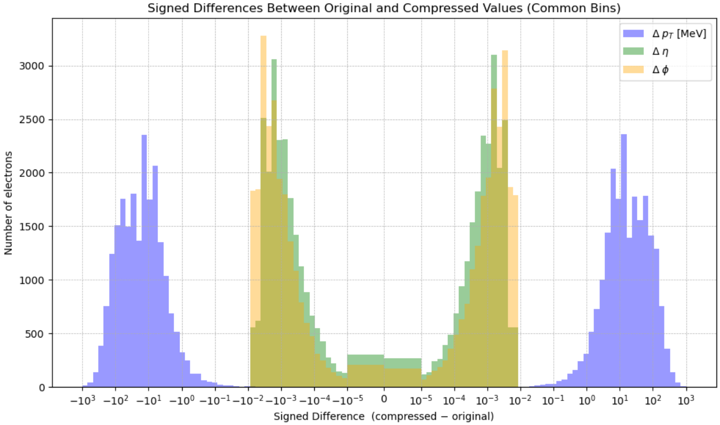


Figure 4. Log-binned signed residuals showing symmetry.

Residual Analysis

Residual Analysis

- Residuals defined as $r = x_{\text{comp}} - x_{\text{orig}}$.
- Histogrammed with symmetric log binning (`symlog`):

    Centered at 0.

    Show clear discrete step patterns.

- 2D log residual mapping:

$$x_0 = \log_{10}\left(|x_{\text{true}}| + 10^{-12}\right), \quad x_1 = \text{sign}\,(r)\,\log_{10}\left(|r| + 10^{-12}\right)$$

- 2D histograms reveal symmetric bands $\Rightarrow$ consistent with quantisation effects.

Residual Analysis

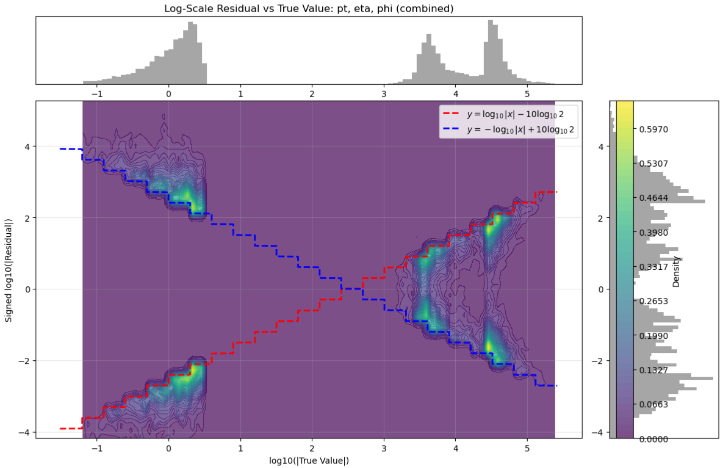


Figure 5. Residuals in log-space vs original values: stepped, banded structure.

Theoretical Work

Theoretical Bounds

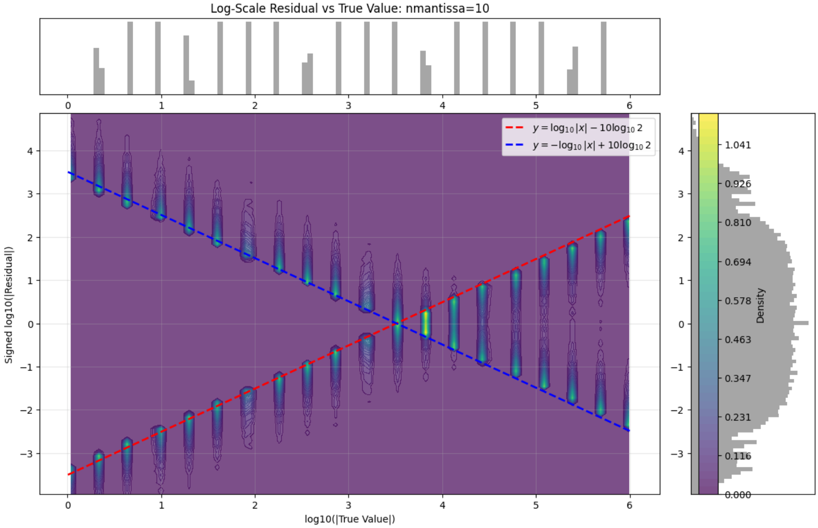


Figure 6. Derived upper and lower bounds for

IEEE 754 Floating Point

- Single-precision float:

$$\text{Value} = (-1)^S \times 1.F \times 2^{E-127}$$

- S: sign bit, E: 8-bit exponent, F: 23-bit mantissa
- Mantissa truncation keeps only the first m bits of F
- Discarded bits cause precision loss

Residuals from Truncation

- Define residual:

$$r = x - \tilde{x}$$

- If discarded bits are zero:

$$r = 0$$

- If discarded bits are non-zero:

$$r > 0 \quad (x > \tilde{x})$$

- Truncation always rounds *toward zero*

Rounding with Mantissa Bits

- Important mantissa bits:

  L: last kept bit
  R: first discarded bit
  T: sticky bits (remaining discarded)

- Rounding rules:

  $$L = 0,\ R = 0 \rightarrow \text{round down}$$
  $$L = 1,\ R = 0 \rightarrow \text{round down}$$
  $$L = 0,\ R = 1 \rightarrow \text{round up}$$
  $$L = 1,\ R = 1 \rightarrow \text{round up}$$

Recovering Rounding Direction?

- Impossible from compressed bits alone
- Multiple original values map to the same truncated pattern
- Rounding direction depends on missing information
- Can only infer probabilistically, not deterministically

Mathematical Formulation

- Floating-point:
$$x = m_x \cdot 2^e, \quad m_x \in [1, 2$$
- Quantized version:
$$\hat{x} = \tilde{m}_x \cdot 2^e, \quad \tilde{m}_x = \mathrm{Trunc}_m\left(m_x\right)$$
- Residual:
$$\Delta = \hat{x} - x = \left(\tilde{m}_x - m_x\right) \cdot 2^e$$
- Quantization step:
$$\left|\tilde{m}_x - m_x\right| < 2^{-m}$$
- Residual bound:
$$|\Delta| < 2^{e-m}$$
- In log scale:
$$\log_{10}|\Delta| < (e - m)\log_{10}(2)$$

Relating Residuals to x

- True value magnitude:

$$\log_{10}|x| \approx e\log_{10}(2)$$

- Substitute:

$$\log_{10}|\Delta| < \log_{10}|x| - m\log_{10}(2)$$

Signed Residual Bounds

- If always round down:

$$\Delta < 0, \quad \log_{10} \Delta = -\log_{10}|x| + m \log_{10}(2)$$

- Final symmetric bound:

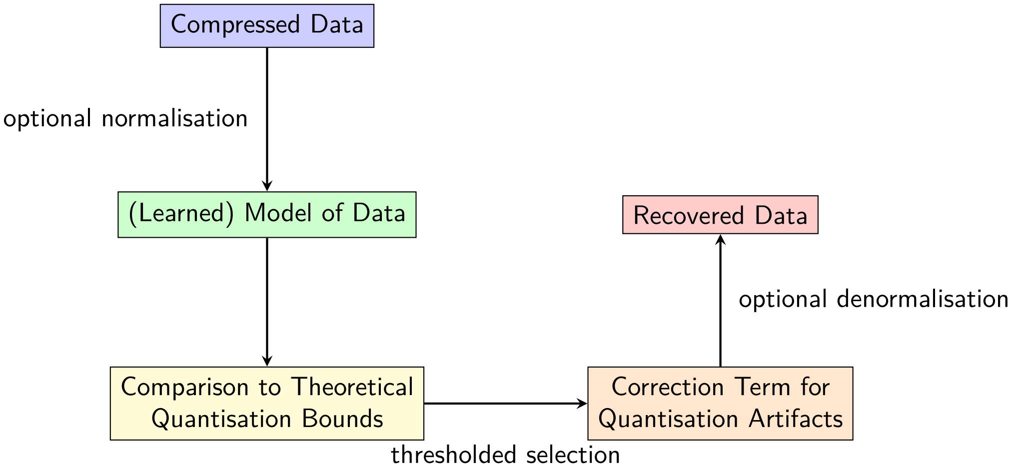
$$\log_{10}|\Delta|_{\text{signed}} \in \left[-\log_{10}|x| + m \log_{10}(2)\,,\ \log_{10}|x| - m \log_{10}(2) \right]$$

Geometric Interpretation

- Bounds form two reflected diagonal lines in $(\log_{10}|x|, \log_{10}|\Delta|)$ space
- Slopes: ± 1
- Intercepts determined by $m \log_{10}(2)$
- Explains visible band structure in residual distributions

Proposed Methods

Proposed System Diagram



Dataset and Preprocessing

- Original and compressed datasets loaded from ROOT files, flattened and masked for finite, positive entries.
- Residuals: $x_{\text{residual}} = x_{\text{true}} - x_{\text{compressed}}$.
- Two dataset modes:

  Sample mode: individual events with residuals.

  Histogram mode: log-spaced histograms per variable, storing $(h_{\text{recon}}, h_{\text{residual}})$.

- Histograms optionally normalized to sum to 1 for KL-divergence-based learning.
- Precomputed mean, std, min/max for normalization and log-binning.

Residual Correction Approaches: Rule-Based

- Compare residuals against theoretical bounds $y_{\text{upper}}, y_{\text{lower}}$ in log space.
- Correction rule:

$$r_{\text{corrected}} = \begin{cases} -r & \text{if } r > 0 \text{ and } |-r - y_{\text{lower}}| < \epsilon, \\ -r & \text{if } r < 0 \text{ and } |y_{\text{upper}} - (-r)| < \epsilon, \\ 0 & \text{otherwise.} \end{cases}$$

- Reconstructed values: $x_{\text{reconstructed}} = x_{\text{compressed}} + r_{\text{corrected}}$.
- Histograms of $x_{\text{reconstructed}}$ closely match the original distribution.

Residual Correction Approaches: Neural Network

- 1D convolutional denoising autoencoder to predict residuals or reconstruct histograms.
- Training strategies:

  MSE Loss: normalized by mean/std.

  KL Divergence Loss: on normalized histograms (sum=1).

- Noisy inputs: Gaussian noise added to compressed values or histograms.
- Evaluation: reconstructed vs original values, histogram overlays, sorted/normalized plots.

Learned Results (Preliminary)

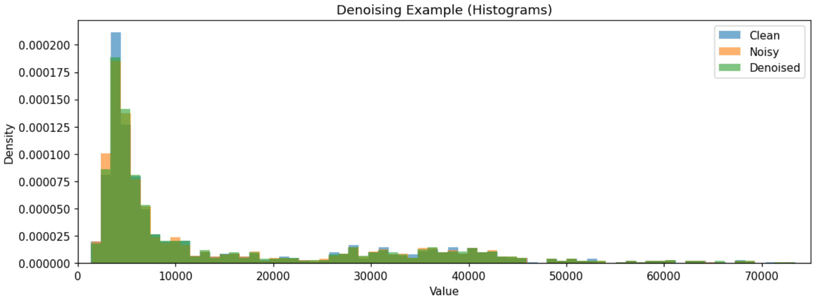


Figure 7. Histogram of Learned Distribution.

Learned Results (Preliminary)

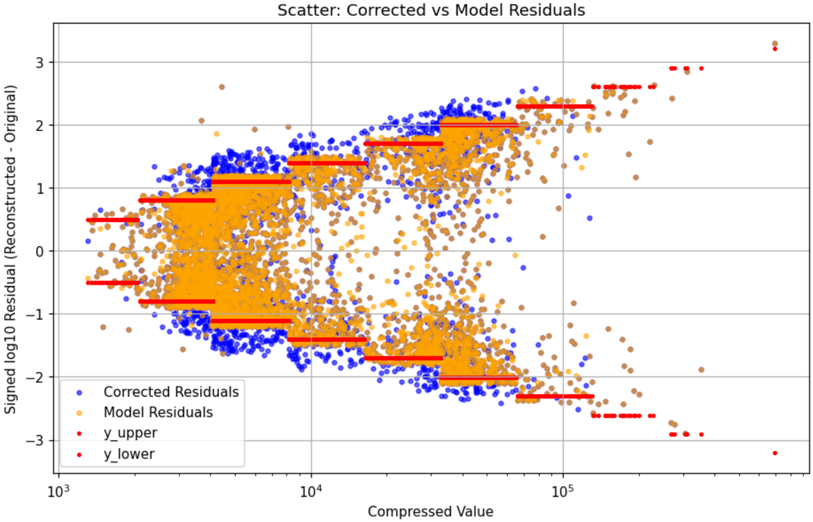


Figure 8. Scatter of error correction.

Proposed Methods

Triaging

Distribution Shift Metrics

- Computed: mean, std, skewness, kurtosis.
- Divergence/distance measures:

  KL/JS divergence, Wasserstein distance, MMD (RBF kernel), KS test, classifier AUC.

- Observations:

  All metrics show extremely small shifts.
  KL/JS divergence $\sim 10^{-4} - 10^{-3}$.
  Normalized MSE $< 10^{-3}$.
  Classifier AUC ≈ 0.5 (distributions indistinguishable).

Distribution metrics over mantissa length

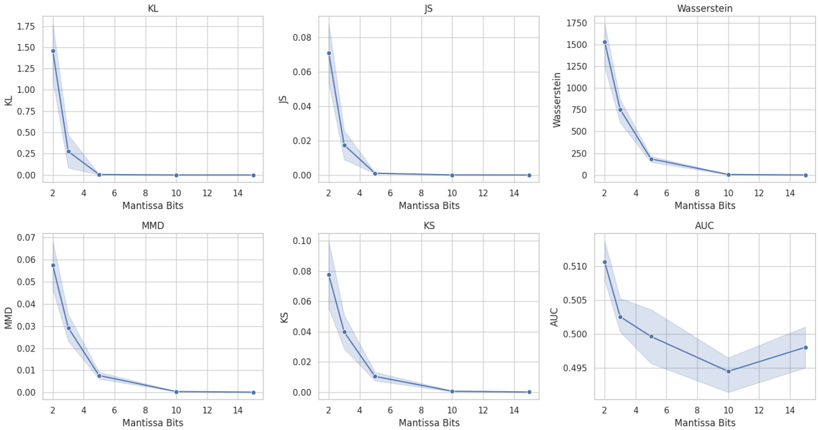


Figure 9. Stepping back to consider the learning goal.

Next Steps

- Currently implementing an 'inpainting' style model which considers that the first x bits of the fraction are known, and therefore the residual must be calculated from the last $23 - x$ bits (for a given exponent, which is also known).

Thanks for your attention!