



Error-bounded Point Cloud Compression Using Truncated Octahedron Quantization

Youyuan Liu*, Longtao Zhang[#], Ruoyu Li[#], Bo Jiang*, Taolue Yang*, Kai Zhao[#], Sheng Di[†], Eduard Dragut*, Sian Jin*

Temple University*, Philadelphia, PA, USA

Florida State University[#], Tallahassee, FL, USA

Argonne National Laboratory[†], Lemont, IL, USA

{youyuan.liu,bo.jiang,taolue.yang,edragut,sian.jin}@temple.edu,{lzhang11,r113m,kai.zhao}@fsu.edu,sdi1@anl.gov

ABSTRACT

With the rapid advancement of large-scale scientific simulations, the massive volume of point cloud data generated has increasingly become a critical bottleneck for scientific storage systems and data management pipelines. Existing point cloud compression techniques integrated into scientific storage systems are designed for sparse geometry and rely on quantization schemes whose optimality assumptions do not hold for dense data. When applied at the compression layer to point clouds, this representation mismatch leads to fundamentally sub-optimal rate-distortion trade-offs that cannot be addressed through parameter tuning or framework-level adaptations. This mismatch increases storage overhead and limits efficient movement and downstream analysis of simulation outputs. This issue arises in scientific data management workflows handling large-scale dense particle datasets. State-of-the-art compression methods fail to fully exploit the redundancies inherent in such data.

We address this limitation by developing a theory of point cloud compressibility for dense data, characterizing fundamental rate-distortion behavior at the representation layer. Guided by this analysis, we introduce XnYZip, an error-bounded lossy compressor based on provably optimal Truncated Octahedron quantization, combined with a locality-aware encoding pipeline using space-filling curves and run-length encoding. Experiments on large-scale scientific datasets demonstrate consistent storage and throughput improvements, achieving up to 3× higher compression ratios, 2.2× faster compression, and 1.2× faster decompression compared to state-of-the-art point cloud compressors under same distortion.

PVLDB Reference Format:

Youyuan Liu, Longtao Zhang, Ruoyu Li, Bo Jiang, Taolue Yang, Kai Zhao, Sheng Di, Eduard Dragut, Sian Jin. Error-bounded Point Cloud Compression Using Truncated Octahedron Quantization. PVLDB, 19(11): 3105 - 3118, 2026.
doi:10.14778/3836663.3836676

PVLDB Artifact Availability:

The source code, data, and/or other artifacts have been made available at <https://github.com/Karshilov/XnYZip>.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 19, No. 11 ISSN 2150-8097.
doi:10.14778/3836663.3836676

1 INTRODUCTION

Data compression has become a fundamental component of modern database and data management systems [6, 8, 23, 28, 40, 62], where it is widely used to reduce storage footprints [31, 55], accelerate queries [7, 71, 72], and mitigate I/O bottlenecks in data-intensive workloads [33]. While built-in lossless compression techniques [31, 55] ensure exact data recovery, they provide limited reduction factors for floating-point scientific datasets. General-purpose lossy compressors and time-series codecs, such as those designed for natural media (e.g., JPEG [35, 61, 65]) or specialized data stores like SummaryStore [3], often lack quantifiable error guarantees, making them unsuitable for scientific workloads that require controlled fidelity and predictable distortion [66]. As a result, error-bounded lossy compression has emerged as a critical storage-layer mechanism in scientific data management systems [1, 15, 32, 62, 67, 68], enabling substantial space savings while preserving analysis-ready representations.

This challenge is amplified by high-performance computing simulations in domains such as molecular dynamics and cosmology, which increasingly generate massive point cloud datasets at unprecedented scale [9, 17, 25, 53]. As exascale applications leverage modern supercomputing platforms to run large ensembles of simulations, they routinely produce dense collections of particles stored at high precision, pushing the memory, storage, and I/O capacities of existing data infrastructures to their limits. For example, the EXAALT project's ParSplice framework executes large numbers of molecular dynamics replicas, each producing thousands to millions of particles per snapshot, resulting in data volumes that quickly reach terabyte to petabyte scale over the course of a single simulation campaign [17]. The sheer magnitude of such point cloud data makes storage, transfer, and post-analysis prohibitively expensive without effective compression support.

Despite the increasing adoption of error-bounded lossy compression in data management systems, existing approaches do not perform uniformly well across different classes of point cloud data. Point clouds generated by scientific simulations are typically much denser than those arising in computer graphics or autonomous driving, where data tends to be relatively sparse. Many current point cloud compressors, including state-of-the-art solutions [22, 56], are primarily designed with such sparse data in mind. Beyond density, these two categories of point clouds also differ in their structural regularity: scan-based point clouds often exhibit a minimum sampling resolution and more regular spatial patterns, while scientific

point cloud data may exhibit either regular or irregular structures depending on the underlying physical model and simulation setup. In contrast, point clouds produced by molecular dynamics and similar simulations are often less constrained by fixed sampling steps and, therefore, display weaker spatial regularity. Thus, compression techniques optimized for sparse and more regular point clouds are often ineffective when applied to dense simulation data with diverse structural characteristics. More fundamentally, *this limitation reflects a mismatch between how compression fidelity is evaluated and how quantization is performed*. The performance of point cloud compressors is predominantly assessed using L2-based metrics such as Mean Squared Error (MSE) [18, 51, 52, 56], yet their underlying quantization schemes almost universally operate by quantizing each coordinate axis independently. This scalar quantization strategy is geometrically equivalent to partitioning space with a cubic lattice, which is provably suboptimal for minimizing MSE [10]. Consequently, *the quantization strategies employed by existing compressors are not optimized, from a geometric perspective, for the very fidelity metrics by which they are judged*.

To address this fundamental mismatch, we develop a theory for the compressibility of dense point cloud data in storage systems. Many existing compressors share a common architecture: distortion is introduced through quantization (or an equivalent mechanism), followed by encoding schemes that exploit correlations across coordinate axes. However, there is currently no unified theory that links these design choices to data compressibility under L2-based fidelity metrics. Our theory fills this gap by deriving compressibility bounds under L2-based MSE, explicitly relating lattice cell geometry to the expected distortion, and yielding density-aware predictions that align with empirical measurements. Guided by—and directly instantiating—our theory, we propose XnYZip, an error-bounded lossy compressor for point cloud data. We replace the traditional cube quantization with **Truncated Octahedron (TO) quantization**, which our theory identifies as optimal for minimizing MSE under vector quantization. XnYZip employs a Space-Filling Curve (SFC) [26, 50] to serialize 3D TO indices into a 1D sequence, preserving spatial locality to enable effective predictive and entropy coding. In addition, an adaptive run-length encoding module is incorporated to exploit high-duplication scenarios common at low bitrates. The main contributions of our work are outlined as follows:

- We develop a theory of dense point cloud compressibility under L2-based fidelity metrics, relating quantization geometry to distortion and rate-distortion behavior.
- Guided by this theory, we design XnYZip, an error-bounded lossy point cloud compressor based on TO quantization and locality-aware encoding.
- We propose an MPI-based distributed compression framework for XnYZip, enabling parallel compression at scale without sacrificing compression ratio and demonstrating favorable iso-efficiency.
- We provide theoretical analyses for the major components of XnYZip, including closed-form derivations and optimality results.
- We evaluate XnYZip on multiple large-scale scientific datasets and demonstrate consistent improvements over state-of-the-art compressors, achieving up to 3× higher compression ratios, 2.2× faster compression, and 1.2× faster decompression under the same distortion levels.

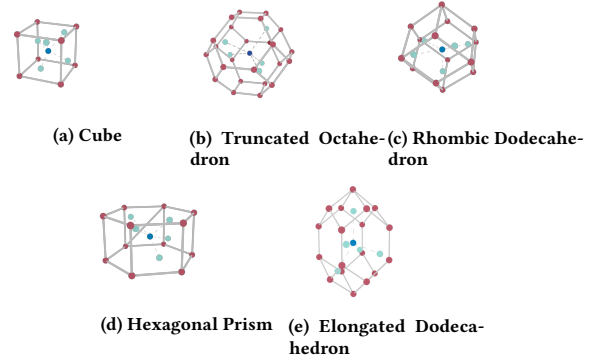


Figure 1: A visual comparison of quantization schemes using the five convex polyhedra that can tile 3D space by translation.

2 BACKGROUND AND RELATED WORK

2.1 Error-Bounded Lossy Compression

Lossy compression provides high compression ratios by accepting controlled information loss, in contrast to lossless methods that preserve data exactly. A critical distinction in this domain is between compressors optimized for perceptual quality, such as JPEG [65], and those designed for numerical or geometric data that provide **strict, user-defined error bounds** [16, 42, 43, 45, 59]. The latter guarantees that the reconstruction error will not exceed a specified threshold, ensures data fidelity in many scientific applications.

Data compression is a core technique in database systems for managing large-scale scientific datasets. Error-bounded lossy compression has emerged as a practical compromise between storage efficiency and analytical fidelity, enabling substantial data reduction while enforcing explicit reconstruction error bounds. Widely used compressors such as SZ [43] and ZFP [45] are effective for structured numerical arrays, while MGARD [4] leverages multiresolution decomposition to support hierarchical error guarantees. In practice, such compressors are commonly deployed as storage-layer primitives in scientific data management pipelines, including database-backed analytics and file-based systems such as HDF5 [62].

However, the effectiveness of these lossy compression frameworks are highly dependent on the structure of the data. Unordered and sparse data such as 3D point clouds lack the regular connectivity that these compressors rely on, requiring specialized algorithms.

2.2 Point Cloud Compression

A 3D point cloud is a set of points in 3D space with geometric and attribute data, whose lack of explicit connectivity poses a fundamental compression challenge [5]. The need for compression is driven by massive data volumes, as large-scale dynamic scenes can easily generate hundreds of millions to billions of points per second in raw acquisition [5].

Existing approaches can be broadly categorized into two main paradigms based on their data organization. The first paradigm, **projection-based compression** [13, 44, 49, 54], maps 3D data into 2D planes to leverage highly optimized video codecs. While exemplified by MPEG’s V-PCC [57] and capable of high compression ratios, these methods inherently rely on orthogonal projections

that partition space into square pixels, often introducing resampling artifacts unsuitable for high-precision scientific analysis. The second paradigm, **direct 3D compression** [24, 29, 37, 38, 41, 60], operates directly in 3D space. This broad category includes methods using tree structures, such as octrees in MPEG G-PCC, or explicit quantization grids to organize spatial data, employing specialized traversals to exploit locality.

For scientific particle and point cloud data, LCP [73] represents the state-of-the-art error-bounded lossy compressor tailored to unordered point sets, employing multi-level reordering strategies to exploit spatial locality. While effective in practice, LCP and related approaches rely primarily on heuristic design choices and do not provide a principled link between quantization geometry and rate-distortion optimality. Existing work on data compressibility has largely focused on general-purpose compression or structured scientific arrays, with limited attention to dense volumetric point clouds or particle data [34]. Hence, current systems offer little theoretical guidance on how quantization geometry influences compression efficiency and distortion behavior in point cloud settings.

Despite the algorithmic differences between these paradigms, virtually all error-bounded lossy point cloud compressors share a fundamental **geometric consensus**: the use of the **cubic lattice** for space partitioning. Whether through recursive octant subdivision or explicit Cartesian quantization, the rate-distortion performance of existing methods is implicitly bounded by the geometric efficiency of the cube. This suggests that the bottleneck lies not only in prediction algorithms, but also in the quantization geometry itself.

The quality of the reconstructed point cloud $\hat{P}$ against the original point cloud P is assessed using several key metrics. **Peak Signal-to-Noise Ratio (PSNR)**, derived from the MSE, is a classic metric for measuring distortion, but it requires a direct point-to-point correspondence. It is defined as:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{Diag}^2}{\frac{1}{N} \sum_{i=1}^N \|p_i - \hat{p}_i\|_2^2} \right),$$

where Diag is the diagonal length of the original point cloud’s bounding box. PSNR is the standard fidelity metric used throughout our evaluation, as the L2-based MSE that it summarizes is the distortion that our error-bounded compressor directly controls.

2.3 Lattice Quantization and Space-Filling

Quantization is the foundational process in lossy compression, which **partitions** the continuous space $\mathbb{R}^n$ into a set of disjoint cells R_i that completely cover the space, i.e., a **tessellation**. Quantization maps all points within a given cell to a single representative value. A particularly principled approach is **lattice quantization**, which maps any given point in $\mathbb{R}^n$ to the nearest node in a lattice, a regular repeating grid of points.

Formally, a lattice L is a discrete set of points generated by all integer linear combinations of a set of basis vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$:

$$L = \left\{ \sum_{i=1}^n k_i \mathbf{v}_i \mid k_i \in \mathbb{Z} \right\}.$$

The lattice results in different space-filling partitions, such as a tessellation by cubes or by Truncated Octahedra (TO), shown in

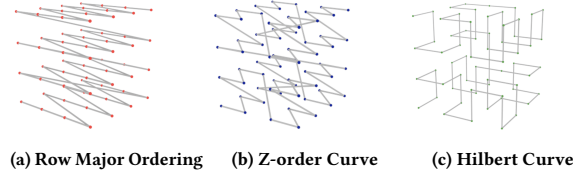


Figure 2: Visual comparison of three traversal paths on a $4 \times 4 \times 4$ grid. Row-major ordering has frequent long jumps, Z-order reduces these jumps, and Hilbert ordering provides stronger spatial locality.

Figure 1. The Voronoi region of a lattice node contains points closer to that node than any other and forms a space-filling polytope.

The problem of determining the most efficient lattice in terms of rate-distortion performance is a classic subject in information theory [10, 48]. Building upon this theoretical foundation, we formalize and extend the analysis to explicitly prove the optimality conditions relevant to our quantization framework, thereby bridging classical lattice theory with practical compressor design.

After lattice quantization, a continuous point cloud is transformed into a sparse set of occupied integer voxel coordinates. To encode these coordinates efficiently, the 3D locations are serialized into a 1D sequence before predictive and entropy coding [14]. A space-filling curve (SFC) provides such a 3D-to-1D ordering while preserving spatial locality [26]: points that are close in 3D space are more likely to remain close in the serialized sequence. Better locality leads to smaller differences between consecutive indexes and therefore a lower-entropy sequence.

Figure 2 compares three representative traversal orders. Row-major ordering is simple but creates large jumps at row and plane boundaries, which hurts compression. Z-ordering [50, 63], improves locality by interleaving coordinate bits, but still contains discontinuities across regions. Hilbert ordering usually provides stronger locality than Z-order [26], reducing large jumps in the serialized sequence. Therefore, we use Hilbert ordering to generate a more compressible first-order difference sequence after quantization.

2.4 Other Related Compression Methods

Several application-aware error-bounded compressors have also been proposed. QPET [33, 47] preserves user-specified Quantities of Interest (QoIs), while TspSZ [69] preserves the topological skeleton of vector fields, including critical points and separatrices. These methods address a complementary problem to ours: they focus on *what* downstream features should be preserved, while our work focuses on *how* quantization geometry affects rate-distortion behavior in dense point cloud data.

Another line of work targets floating-point columns in data management systems. ALP [2] provides high-throughput lossless compression for tabular floating-point columns, and BUFF [46] supports SIMD-friendly queries under user-specified bit precision. A recent survey [27] studies this regime in detail. Although effective for one-dimensional numeric streams, these methods are less effective for 3D point clouds because they encode axes independently and do not exploit spatial structure.

Vector compression, such as Product Quantization (PQ) [36] and RabbitQ [20], are mainly designed for high-dimensional embeddings

Table 1: List of Symbols and Abbreviations

Symbol	Meaning	Symbol	Meaning
P	Point Set	$G(\cdot)$	Dimensionless second moment
V	Volume	N	Number of points
$Enc(\cdot)$	Encoder	$Dec(\cdot)$	Decoder
d	Distance	$Diag$	Diagonal length of point cloud
Λ	Voxel Set	p	A single point
λ	A single voxel	$\mathcal{E}$	Total square error
$\mathcal{P}$	A polyhedron	M	Number of voxel
S	Symbol	H	Entropy
I	Mutual information	c	Centroid point
$E(\cdot)$	Error	$\mathcal{R}$	Partition/Region
Q	Quantizer	$D(\cdot, \cdot)$	Distortion
Abbreviation	Explanation		
TO	Truncated Octahedron		
MSE	Mean Squared Error		
RLE	Run-length Encoding		
SFC	Space-Filling Curve		
PSNR	Peak Signal-to-Noise Ratio		

and approximate nearest-neighbor search. They prioritize retrieval accuracy rather than reconstruction fidelity, and their coarse quantization is not suitable for scientific and industrial point clouds that require strict geometric error bounds. Therefore, our work complements these methods by providing a quantization-geometry view of error-bounded point cloud compression.

3 PROBLEM FORMULATION

We formulate point cloud compression as a 3D quantization problem and use this formulation to motivate our choice of the BCC lattice and the resulting TO quantization.

Problem Definition. The objective of our project is to develop a lossy compressor for point cloud data that is **natively optimized for the geometric fidelity metrics** commonly used in the field. The fundamental problem is to find an encoder Enc and a decoder Dec that minimize the rate R (in bits per point) for a given target distortion D_{target} . We focus on the fixed-distortion setting because scientific post-analysis typically requires a user-defined error guarantee, whereas traditional media compression often fixes the bitrate. The input is a set of 3D points, $P = \mathbf{p}_1, \dots, \mathbf{p}_N$ where $\mathbf{p} * i \in \mathbb{R}^3$. The encoder Enc maps P to a compressed bitstream of length $|Enc(P)|$, and the decoder Dec recovers the *reconstructed* point cloud $P' = Dec(Enc(P))$. The rate R is a function of $|Enc(P)|$ and the number of points N , while the distortion $D(P, P')$ measures the dissimilarity between P and its reconstruction P' ; the error bound is enforced on this reconstruction, not on the bitstream, which is why P' must appear in the formulation. The optimization problem is formulated as:

$$\min R = \frac{|Enc(P)|}{N} \quad \text{s.t.} \quad D(P, P') \leq D * target \quad (1)$$

Quantization as the Loss Model. Transform-based methods, such as those derived from the Discrete Fourier Transform (DFT) or Discrete Cosine Transform (DCT), have proven effective for error-bounded compression on structured data like images or regular grids [12, 35, 45]. In contrast, a general point cloud is an *unordered set of coordinates*, where point indices are arbitrary and lack physical

meaning. Consequently, applying a DFT/DCT to such an index-based sequence yields a spectrum that is not an invariant property of the underlying geometry.

Quantization is the cornerstone of error-bounded lossy compression for point clouds. It partitions the continuous space $\mathbb{R}^3$ into a finite set of disjoint voxels V_i and maps all points within a voxel to a single representative point c_i . Crucially, this process inherently provides an error-bounding mechanism: the maximum possible error for any point is geometrically constrained by the dimensions of its corresponding voxel, e.g., its covering radius. *The problem of designing an optimal error-bounded compressor for 3D point cloud is therefore transformed to finding the optimal quantization lattice.*

Linking Geometry, Distortion, and Rate. The choice of quantization lattice governs the trade-off between distortion D and rate R .

Formalizing the Quantized Representation. After quantization, the original point cloud P is transformed into a multiset of N voxel identifiers, $\Lambda = \lambda_1, \lambda_2, \dots, \lambda_N$. This multiset is not an Independent and Identically Distributed (i.i.d.) source; it carries significant **spatial redundancy**: the geometric coherence of the underlying point cloud introduces strong statistical dependence among neighboring voxel identifiers.

Formalizing Rate via a Two-Part Information Model. The theoretical minimum rate R is determined by the true entropy of this structured source. We model the total information content using two components: **structural information**, which specifies *which* M of the N_{total} possible voxel locations are occupied, and **statistical information**, which describes the frequency distribution of the N points among these M occupied locations.

Under a fixed distortion constraint, a smaller number of occupied voxels M reduces the number of possible spatial configurations and thus lowers the structural complexity. It also typically leads to a more skewed frequency distribution for the N points, which lowers the statistical information. Assuming a locality-preserving encoding that approaches the entropy bound, minimizing the number of occupied voxels M yields a lower achievable rate R : $\min(M) \implies \min(R)$

The Optimal Quantization Problem. For a given target distortion D_{target} , a more efficient quantization geometry will require a smaller number of voxels, M , to represent the point cloud. Based on the derivation above, a smaller M results in a representation with lower total information content and thus a lower theoretical compressed bitrate. Therefore, *the problem of finding the optimal quantizer is formalized as finding the lattice that minimizes the number of occupied voxels M for a given target distortion.*

The Encoding Problem: From Geometry to Information. The quantization stage solves the geometric representation problem, transforming the continuous point cloud into a discrete set of occupied voxel locations. The subsequent problem is to losslessly encode this set by exploiting its inherent redundancies. This requires addressing two types of redundancy: **statistical redundancy**, which comes from the non-uniform frequency distribution of the symbols, and **spatial redundancy**, which comes from the statistically dependent correlation of voxel locations in space.

Quantifying Spatial Information. We define the **Spatial Information** within a sequence of voxel locations S as the degree to which it deviates from being i.i.d. This can be quantified by the **Mutual Information** between adjacent elements in the sequence:

$$I(S_i; S_{i-1}) = H(S_i) - H(S_i|S_{i-1}) \quad (2)$$

where $H(S_i)$ is the zeroth-order entropy and $H(S_i|S_{i-1})$ is the first-order conditional entropy. A high mutual information indicates strong adjacent-symbol dependence and predictability, while a value near zero indicates that the sequence is approximately i.i.d. In our encoding pipeline, locality-preserving ordering and delta coding reduce this dependence and convert much of the spatial redundancy into a lower-entropy residual stream, which can then be compressed by entropy coding.

3.1 Quantifying the Compressibility

While the TO lattice offers a theoretical geometric advantage, realizing this gain in practice depends on the data distribution relative to the quantization scale. To characterize the regime where our method outperforms traditional approaches, we introduce two analysis metrics: *Point Density* and *Distribution Deviation*.

(1) Point Density (ρ). This metric measures the average occupancy of the quantization cells. For a given error bound ϵ , let M_{occ} be the number of non-empty voxels. The density is defined as:

$$\rho(\epsilon) = \frac{N}{M_{occ}(\epsilon)} \quad (3)$$

The geometric packing advantage of the TO lattice is realized primarily in the **Dense Regime** ($\rho \gg 1$), where multiple points cluster into single voxels. If $\rho \rightarrow 1$, quantization degrades to coordinate enumeration, and the voxel geometry becomes less significant.

(2) Distribution Deviation (δ). The theoretical optimality of lattice quantization relies on a **Spatial Uniformity Assumption**, where points are assumed to be uniformly distributed within a voxel, as commonly used in high-resolution quantization theory [10, 48]. Under this assumption, the L_2 error follows a theoretical distribution Q_{sim} dictated by the voxel geometry.

To quantify the validity of this assumption on real-world datasets, we measure the divergence between the *empirical* L_2 error distribution (P_{emp}) and the *theoretical* distribution (Q_{sim}) using the **Kullback-Leibler (KL) Divergence**:

$$\delta(\epsilon) = D_{KL}(P_{emp}||Q_{sim}) = \sum_{k=1}^K p_k \log \frac{p_k}{q_k} \quad (4)$$

$\delta \approx 0$ indicates good agreement with the uniformity assumption. A large δ indicates strong structural bias or boundary effects that violate spatial uniformity, reducing the geometric advantage.

4 METHODOLOGY

In this section, we introduce our proposed framework, XnYZip, shown in Fig. 3. We then establish the theoretical foundation for our choice of quantization geometry through a series of lemmas.

4.1 Optimal Quantization Geometry

4.1.1 Assumption. Proceeding with the **Spatial Uniformity Assumption** established in Section 3.1, we focus our analysis on the high-resolution regime. This regime is directly aligned with

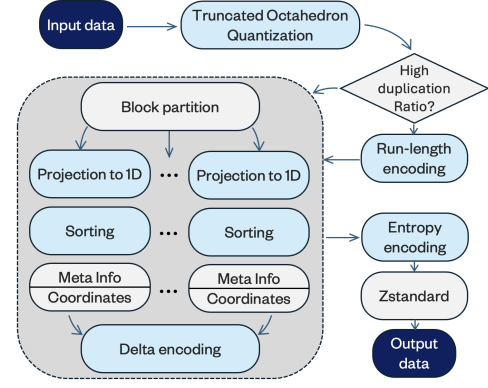


Figure 3: The overview of XnYZip.

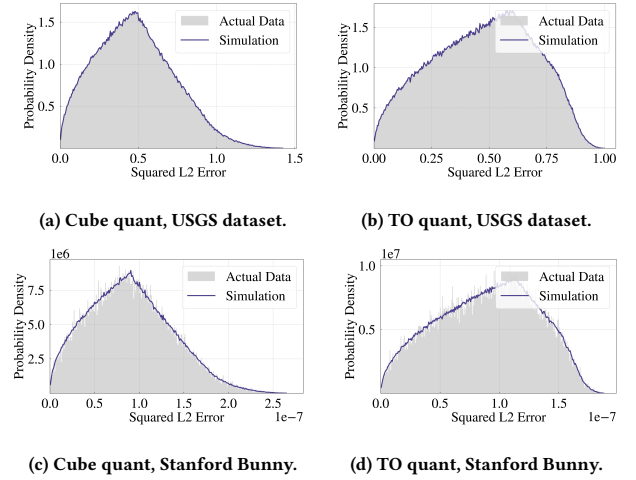


Figure 4: Comparison of the actual quantization error distributions (histograms) against the theoretical error distributions (Solid blue line). The comparison is shown for both the Cube and Truncated Octahedron (TO) quantizers on the dense USGS dataset (top row) and the sparse Stanford Bunny dataset (bottom row), with reasonable eb .

the requirements of scientific simulations, high-accuracy numerical modeling, and high-fidelity reconstruction, where quantization errors are expected to be much smaller than the intrinsic spatial variations of the data, making the local point distribution within each voxel well approximated as uniform. We empirically validate this assumption by our experiments shown in Figure 4, where the actual error distributions closely match the simulations derived from this uniform model. Then we prove that the search for an optimal quantization voxel under the MSE distortion metric is fundamentally restricted to the set of convex polyhedrons.

Lemma 4.1. Let a quantizer achieve the global minimum MSE. Then each of its quantization voxels is a convex polyhedron.

Proof. Let a quantizer Q be defined by a set of centers $\{c_i\}$ and a corresponding partition of space into voxels $\{\lambda_i\}$. The total MSE is $E(Q) = \sum_i \int_{\lambda_i} \|\mathbf{x} - c_i\|^2 d\mathbf{x}$. For $E(Q)$ to be minimal for a given

set of centers, the partition $\{\mathcal{R}_i\}$ must follow the nearest-neighbor rule, which means the closest center of a point p_i is the center of the region R that p_i belongs to. We prove this by contradiction.

Let us assume that an optimal partition $\{\mathcal{R}_i^*\}$ for a given set of centers $\{\mathbf{c}_i^*\}$ is not a nearest-neighbor partition. Formally, this corresponds to:

$$\begin{aligned} & \exists \mathbf{p} \in \mathcal{R}_i^* \text{ and } j \neq i \text{ such that } \|\mathbf{p} - \mathbf{c}_j^*\|^2 < \|\mathbf{p} - \mathbf{c}_i^*\|^2. \\ \Rightarrow & \text{Construct } \{\mathcal{R}_k'\} \text{ by reassigning } \mathbf{p} \text{ from } \mathcal{R}_i^* \text{ to } \mathcal{R}_j^*. \\ \Rightarrow & \text{The change in total MSE is } \Delta E = \|\mathbf{p} - \mathbf{c}_j^*\|^2 - \|\mathbf{p} - \mathbf{c}_i^*\|^2 < 0. \end{aligned}$$

The existence of a partition with a lower total MSE contradicts the assumption that $\{\mathcal{R}_i^*\}$ is optimal. Thus, any MSE-optimal partition must be a nearest-neighbor partition.

The regions included by this nearest-neighbor rule can now be analyzed. For any two distinct centers, $\mathbf{c}_i^*$ and $\mathbf{c}_j^*$, the boundary of points equidistant from both is a hyperplane. The set of all points closer to $\mathbf{c}_i^*$ is formally defined by the inequality $\|\mathbf{p} - \mathbf{c}_i^*\| \leq \|\mathbf{p} - \mathbf{c}_j^*\|$, which describes a closed half-space. Since a half-space is a convex set, the quantization voxel λ_i^* can be defined as the set of all points closer to $\mathbf{c}_i^*$ than to any other center, which is therefore the intersection of multiple such half-spaces (one for each $j \neq i$). Consequently, each voxel λ_i^* is necessarily a convex polyhedron as the intersection of any number of convex sets is a convex set. $\square$

Lemma 4.1 states that the optimal quantization voxel must be a convex polyhedron. Whose representative point that minimizes the internal MSE is its centroid [21]. We then formally define a criterion for comparing the efficiency of different voxel geometries.

Lemma 4.2. Consider two lattice quantizers, Q_A and Q_B , generated by space-filling convex polyhedrons [10] $\mathcal{P}_A$ and $\mathcal{P}_B$ respectively. At a constant rate (i.e., when their voxel volumes are equal), Q_A achieves a lower MSE than Q_B if and only if the dimensionless second moment of its voxel is lower, i.e., $G(\mathcal{P}_A) < G(\mathcal{P}_B)$.

Proof. Let MSE_A and MSE_B denote the MSE of the quantizers Q_A and Q_B , respectively. The overall MSE is proportional to the unnormalized MSE within a single quantization voxel, $\mathcal{E}(\lambda)$:

$$E(\lambda) = \frac{1}{V(\lambda)} \int_{\lambda} \|\mathbf{x} - \hat{\mathbf{x}}\|^2 d\mathbf{x}$$

Suppose the shape of voxel λ is a polyhedron $\mathcal{P}_\lambda$. The dimensionless second moment, $G(\mathcal{P})$, is a scale-invariant measure of the voxel's shape efficiency, defined as:

$$G(\mathcal{P}_\lambda) = \frac{\int_{\lambda} \|\mathbf{x} - \hat{\mathbf{x}}\|^2 d\mathbf{x}}{[V(\lambda)]^{\frac{n+2}{n}}}$$

By substituting the definition of $E(\lambda)$ into the formula for $G(\mathcal{P}_\lambda)$, we can express the voxel's MSE as a direct function of its shape and volume efficiency: $E(\lambda) = G(\mathcal{P}_\lambda) \cdot [V(\mathcal{P}_\lambda)]^{2/n}$.

Lemma 4.2 states that both quantizers operate at a constant rate, which implies their voxel volumes are equal. Let $V = V(\lambda_A) =$

$V(\lambda_B)$. We can now establish a chain of logical equivalences:

$$\begin{aligned} & Q_A \text{ achieves a lower MSE than } Q_B \\ \Leftrightarrow & MSE_A < MSE_B \\ \Leftrightarrow & E(\lambda_A) < E(\lambda_B) & (\text{Overall MSE} = \text{voxel MSE}) \\ \Leftrightarrow & G(\mathcal{P}_{\lambda_A}) [V(\lambda_A)]^{2/n} & \\ & < G(\mathcal{P}_{\lambda_B}) [V(\lambda_B)]^{2/n} & (\text{By MSE relation}) \\ \Leftrightarrow & G(\mathcal{P}_{\lambda_A}) V^{2/n} < G(\mathcal{P}_{\lambda_B}) V^{2/n} & (\text{Premise: } V(\lambda_A) = V(\lambda_B)) \\ \Leftrightarrow & G(\mathcal{P}_{\lambda_A}) < G(\mathcal{P}_{\lambda_B}) & (\text{const. } V^{2/n} > 0) \end{aligned}$$

This directly shows that a lower MSE is equivalent to a lower dimensionless second moment, given equal voxel volumes. $\square$

Lemma 4.2 provides a definitive metric, $G(P)$, for evaluating the quantization efficiency of any given convex polyhedron. We now address whether an optimal tiling should be composed of a single uniform shape or a composite of multiple different shapes.

Lemma 4.3. To minimize L2-based MSE, a uniform tiling composed of a single optimal polyhedron is strictly superior to any composite tiling of varying shapes, orientations, or volumes.

Proof. We compare two scenarios for tiling a given space. First, consider a composite tiling of N voxels with potentially varying shapes $\{S_i\}$ and volumes $\{V_i\}$. Its total squared error is $\mathcal{E}_{\text{comp}} = \sum_{i=1}^N G(S_i) V_i^{5/3}$. Second, consider a uniform tiling of N voxels, where each voxel has the optimal shape (i.e., with shape constant G_{opt}) and the same average volume $V_{\text{avg}} = (\sum V_i)/N$. Its total error is $\mathcal{E}_{\text{uni}} = N \cdot G_{\text{opt}} \cdot (V_{\text{avg}})^{5/3}$.

We prove $\mathcal{E}_{\text{comp}} \geq \mathcal{E}_{\text{uni}}$ in two steps. By definition, the shape constant of any voxel, $G(S_i)$, cannot be better than that of the optimal shape, G_{opt} , thus $G(S_i) \geq G_{\text{opt}}$. This yields the first inequality:

$$\mathcal{E}_{\text{comp}} = \sum_{i=1}^N G(S_i) V_i^{5/3} \geq \sum_{i=1}^N G_{\text{opt}} V_i^{5/3} = G_{\text{opt}} \sum_{i=1}^N V_i^{5/3}$$

Next, we apply **Jensen's inequality** to the convex function $f(V) = V^{5/3}$, which states that the average of the function's values is greater than or equal to the function of the average value. This gives the second inequality:

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N V_i^{5/3} & \geq \left(\frac{\sum_{i=1}^N V_i}{N} \right)^{5/3} = (V_{\text{avg}})^{5/3} \Rightarrow \\ \sum_{i=1}^N V_i^{5/3} & \geq N \cdot (V_{\text{avg}})^{5/3} \Rightarrow \sum_{i=1}^N G_{\text{opt}} V_i^{5/3} \geq N \cdot G_{\text{opt}} \cdot (V_{\text{avg}})^{5/3} \end{aligned}$$

Combining these steps, we find that $\mathcal{E}_{\text{comp}} \geq \mathcal{E}_{\text{uni}}$. This inequality is strict in any practical composite tiling, as it will either contain suboptimal shapes (where $G(S_i) > G_{\text{opt}}$), including rotated versions of an optimal shape, or non-uniform volumes ($V_i \neq V_{\text{avg}}$). A uniform tiling with the single optimal shape is therefore provably the most efficient solution. $\square$

Lemma 4.1–4.3 provides the formal underpinnings for narrowing the search for an MSE-optimal quantizer to a single, uniformly-tiled, space-filling convex polyhedron that minimizes the dimensionless second moment, $G(P)$. The solution to this problem is found by

Table 2: Comparison of dimensionless second moment $G(P)$ for various three-dimensional polyhedra $\mathcal{P}$

P	Tetrahedron	Cube*	Hexagonal prism*	Rhombic dodecahedron*	Truncated octahedron*	Dodecahedron	Icosahedron	Sphere
$G(P)$	0.1040042 ...	0.0833333 ...	0.0812227 ...	0.0787451 ...	0.0785433 ...	0.0781285 ...	0.0778185 ...	0.0769670 ...

* A space-filling polyhedron.

Table 3: Comprehensive comparison of quantization efficiency for various space-filling polyhedra under different metrics, using the Cube as a baseline. Bold indicates the best choices.

Polyhedron ($\mathcal{P}$)	Intrinsic Efficiency $G(\mathcal{P})^a$	Constant MSE/PSNR Volume $\uparrow^b$ Voxel $\downarrow^c$	Constant Hausdorff Efficiency $\uparrow^d$
Cube (Baseline)	0.08333	1.0x 0.0%	1.0x
Hexagonal prism	0.08122	1.039x 3.7%	1.54x ^e
Truncated octahedron	0.07854	1.092x 8.4%	1.77x
Rhombic dodecahedron	0.07875	1.088x 8.1%	1.88x

^a Dimensionless Second Moment ($G(\mathcal{P})$): The intrinsic measure of MSE efficiency. Lower is better.

^b Volume Gain: Relative voxel volume for the same MSE. Higher is better.

^c Voxel Reduction: Percentage reduction in total voxels for the same total MSE. Higher is better.

^d Hausdorff Efficiency Gain: A proxy for performance under a Hausdorff distance constraint, based on the voxel's covering efficiency. Higher is better.

^e Note on Hexagonal Prism: Unlike the other polyhedra, the geometry of a hexagonal prism depends on its height-to-width ratio. The value shown corresponds to the theoretical maximum efficiency, achieved at an optimal ratio of $h/a = \sqrt{2}$.

consulting the established values of $G(P)$ for various polyhedra in the classical 3D space-filling theory [10].

Table 2 shows that the Truncated Octahedron (TO), the Voronoi voxel of the Body-Centered Cubic (BCC) lattice, has the lowest known normalized second moment among 3D space-filling candidates, with $G(P) \approx 0.07854$. As quantified in Table 3, replacing the cubic lattice, for which $G(P) \approx 0.08333$, with the BCC lattice achieves the same MSE using **8.4% fewer voxels**. This reduction improves rate-distortion performance because fewer voxel indices must be encoded at the same reconstruction quality. We therefore use the BCC lattice as the basis of our quantization strategy.

This geometric advantage is useful only if quantization remains computationally efficient. Although the truncated octahedron is more complex than a cube, the following section presents a fast algebraic BCC quantization algorithm with complexity comparable to standard cubic-lattice quantization.

4.2 Efficient TO Quantization

Cube quantization is intuitive with low computational complexity, as mapping a point to a cubic lattice only requires a component-wise floor or round operation. At first glance, adopting a more geometrically complex quantizer like the Truncated Octahedron seems to compromise this efficiency. However, this is not the case. The quantization can be simplified by focusing on the spatial arrangement of the voxel centers, which form a BCC lattice. For a BCC lattice aligned with an integer grid, the coordinates (x, y, z) of any lattice point satisfy a simple parity property: **the components are either all even or all odd**, as shown in Figure 5. This property allows for a direct, analytical mapping from a continuous coordinate to a discrete lattice index.

Our BCC quantization implementation has constant-time computation comparable to cube quantization. Given a point $\mathbf{p}$ and

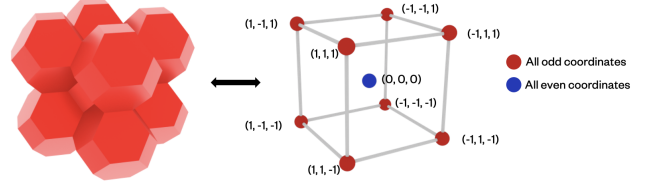


Figure 5: The centers of truncated octahedra form a BCC layout. Each center point is constructed using either all even or all odd coordinates in the normalized BCC coordinate system.

L2 error bound ϵ , it first normalizes the point by the BCC lattice scale $2\epsilon/\sqrt{5}$. It then locates the unit cube containing the normalized point and identifies the two BCC lattice candidates in that cube: the all-even and all-odd integer vertices. The quantized coordinate is obtained by selecting the candidate with the smaller Euclidean distance to the normalized point. For example, when $\lfloor \mathbf{p}' \rfloor = (0, 1, 2)$, the local cube contains two valid BCC lattice candidates: the all-even point $\mathbf{q}_E = (0, 2, 2)$ and the all-odd point $\mathbf{q}_O = (1, 1, 3)$. This procedure requires a few fixed operations, yielding $\Theta(1)$ complexity while exploiting the periodic structure of the BCC lattice.

4.3 Encoding

After quantization, the continuous point cloud is transformed into a set of discrete integer coordinates. The remaining task is to losslessly encode these coordinates by exploiting two types of redundancy: inter-axis correlation among (x, y, z) coordinates and spatial locality among nearby occupied cells.

Joint Coordinate Encoding. A straightforward strategy is to encode the x , y , and z coordinates as three independent streams. However, for geometric data, these coordinates are inherently correlated. From an information-theoretic perspective, joint encoding is more efficient because the joint entropy $H(X, Y, Z)$ is no larger than the sum of marginal entropies $H(X) + H(Y) + H(Z)$ [11]. Therefore, instead of compressing each axis independently, XnYZip first maps each 3D voxel coordinate to a single 1D index and then encodes the resulting index sequence.

Serialization via Space-Filling Curves. The set of quantized voxel coordinates is unordered. To exploit spatial locality, we serialize the 3D coordinates into a 1D sequence using a space-filling curve (SFC). A simple row-major order is easy to compute, but it creates large artificial jumps at row and plane boundaries, as shown in Figure 2. Block-wise row-major indexing, as used in compressors such as LCP, reduces the dynamic range by converting global coordinates into local coordinates, but it does not fully remove the long-tail behavior caused by row and plane transitions.

To reduce these jumps, we use locality-preserving SFCs for the 3D-to-1D projection. SFC ordering keeps nearby 3D cells close in the serialized sequence, producing smaller consecutive differences

and a more compressible stream. In the evaluation, we denote the Z-order and Hilbert variants of our framework as XnYZip-z and XnYZip-h, respectively.

Delta Encoding. After SFC serialization, we obtain an ordered integer sequence S . The output of quantization is a **sparse set of discrete integer coordinates**, rather than a dense scalar field where every grid location has a value. Therefore, interpolation- or stencil-based predictors, such as Lorenzo-style predictors, are not suitable because they assume regular neighboring samples exist.

Instead, the most reliable local correlation comes from the 1D adjacency induced by SFC ordering. We therefore use a first-order predictor where the prediction for S_i is its immediate predecessor S_{i-1} . The residual is encoded by delta encoding: Delta encoding is widely used in prediction-based scientific data compressors [43, 73]. We adopt it here because it relies only on the local adjacency induced by SFC ordering and does not assume an underlying grid structure or functional smoothness required by interpolation- or stencil-based predictors.

Optional Optimization: Run-Length Encoding. The output of quantization is an unordered multiset of N integer coordinates. In dense datasets, multiple points may be quantized to the same coordinate, creating many duplicates. To exploit this redundancy, XnYZip optionally applies **run-length encoding (RLE)** immediately after quantization. RLE transforms the multiset of N points into unique coordinate-count pairs, allowing the later stages to operate on a smaller set of occupied voxel locations while encoding the count stream separately.

Entropy and Dictionary Coding. After RLE, SFC serialization, and delta encoding, the remaining residual stream mainly contains statistical redundancy. We use **Huffman coding** to assign shorter codes to more frequent residual symbols. Finally, the resulting byte stream is passed to **Zstd** to capture remaining byte-level patterns and improve the overall compression ratio.

Worked Example: Encoding a 6-Cell Block. To make the bit budget concrete, consider a 13-point local block whose BCC quantization stage emits 6 unique cells with the integer coordinates and per-cell counts below. Each cell coordinate is either all-even or all-odd, as required by the BCC lattice.

cell (x, y, z)	count	Z-order index	delta
(0, 0, 0)	3	0	0
(1, 1, 1)	4	7	7
(3, 3, 1)	1	15	8
(0, 0, 2)	1	32	17
(1, 1, 3)	2	39	7
(2, 2, 0)	2	48	9

The RLE stage decomposes the 13-point multiset into the cell-index column and the count column above; the two streams are encoded independently from this point on. Sorting cells by their Z-order index produces a monotone sequence whose first-order differences (delta column) are bounded by the local cell spacing rather than by the bbox diagonal.

On the delta stream [0, 7, 8, 17, 7, 9] the only repeated symbol is 7 (frequency 2); a Huffman code that assigns 7→0, 0→100, 8→101, 17→110, 9→111 encodes the six symbols in 3+1+3+3+1+3 = 14 bits. On the count stream [3, 4, 1, 1, 2, 2], with frequencies {1:2, 2:2, 3:1, 4:1}, a Huffman code 2→0, 1→10, 3→110, 4→111 uses 12 bits. The whole block is therefore encoded in 14 + 12 = 26 bits of

payload, compared with 54 bits for the raw cell-and-count list (6×6-bit Morton codes plus 6×3-bit counts) and ≈ 78 bits if the 13 original points were written as integer triples. The downstream Zstd stage further compresses the byte-level structure of the Huffman output.

4.4 Distributed Compression

The single-node pipeline in Section 4.3 assumes that the input point cloud can be processed within the memory of one machine. This assumption is often violated by production scientific workloads. For example, a single HACC particle file contains on the order of 10^9 points (about 12.9 GB), and a full simulation snapshot is partitioned into hundreds of such files. To support such workloads, we extend XnYZip to an MPI-based map-shuffle-reduce pipeline over the quantized cell table, as shown in Algorithm 1.

Each rank reads a disjoint slice of the input, computes the global bounding box using one MPI_Allreduce, and quantizes its local points into a cell-to-count hash map (Map). The ranks then route cell-count entries according to their corresponding SFC key ranges using MPI_Alltoallv (Shuffle). After the shuffle, each rank owns a disjoint contiguous SFC range, aggregates duplicate cells received from different ranks (Reduce), encodes the owned cells using Section 4.3, and writes a self-contained part file. This representation preserves the same global cell-count table as the monolithic compressor: each occupied cell appears in exactly one rank’s output. The only extra storage comes from per-rank headers and part boundaries. Moreover, the shuffle operates on unique cell-count pairs rather than raw points, so dense point clouds often incur much lower communication volume than their input size.

Algorithm 1: Distributed Compression via MPI Map-Shuffle-Reduce

Input : Point cloud file F , error bound ϵ , rank r out of R ranks
Output : Compressed file.

```

1  $slice_r \leftarrow$  byte range of  $F$  assigned to rank  $r$ ; (min, max)  $\leftarrow$ 
   MPI_Allreduce(per-axis extrema of  $slice_r$ );  $local\_table \leftarrow$  empty hash
   map;
2 foreach point  $p \in slice_r$  do
3    $cell \leftarrow$  BCCQuantize( $p, \epsilon$ );  $local\_table[cell] += 1$ ;
4 end
5 foreach ( $cell, count$ )  $\in local\_table$  do
6    $key \leftarrow$  SFC( $cell$ ); append ( $cell, count$ ) to  $sendbuf[OWNER(key)]$ ;
7 end
8  $recvbuf \leftarrow$  MPI_Alltoallv( $sendbuf$ );  $owned \leftarrow$  aggregate counts in
    $recvbuf$  by  $cell$ ;  $bytes \leftarrow$  Encoding pipeline of S 4.3 applied to  $owned$ ;
```

A fixed-bit-width Morton partition can assign SFC ranges to ranks with low overhead, but it may suffer from load imbalance when the bounding box is highly skewed. For example, on the HACC volume of $32 \times 64 \times 256$, the longest axis dominates the leading Morton bits and can concentrate many cells onto a few ranks. We therefore use sample-sort splitters: each rank samples local cells, rank 0 selects approximate (i/R) -th quantile splitters, and the splitters are broadcast to all ranks. The encoding pipeline is unchanged, but the SFC ranges are chosen to balance the number of owned cells across ranks.

This design follows the parallelization profile of the single-node pipeline. TO quantization and local hash-table construction are embarrassingly parallel, and each rank performs local sorting and encoding independently after the shuffle. Since point clouds are

Table 4: Benchmark datasets used in the evaluation. *Size* denotes the size of the upstream corpus from which each dataset is drawn; *Tested Size* and *Points* denote the data actually compressed in our experiments, which for terabyte-scale corpora corresponds to representative subsets of the full archive.

Dataset	Domain	Size	Tested Size	Points	$\rho^{\ddagger}$
YIIP [30]	MD*	4 GB	1.3 MB	1.12×10^5	1.00
Stanford Dragon [58]	CG*	43 MB	5.0 MB	4.38×10^5	1.33
GroEL-GroES [70]	Biology	5 MB	0.7 MB	5.87×10^4	1.00
Vesicles [39]	Biology	807 MB	20.0 MB	1.75×10^6	1.00
USGS [64]	Geology	>200 TB	175 MB	1.53×10^7	10.20
EXAALT [17]	MD*	4 GB	32.8 MB	2.87×10^6	1.00
HACC [25]	Cosmology	>200 TB	12.9 GB [†]	1.07×10^9 [†]	5.39
WarpX [19]	Plasma	8 TB	2.76 GB	2.41×10^8	1.01

* MD for Molecular Dynamics, CG for Computer Graphics.

[†] A smaller subset (~3.13 GB) has been used for compression ratio and case study evaluation. Larger fullset is used for distributed compression experiments.

[‡] Point density $\rho = N/M_{occ}$ measured at $eb = 10^{-3}$

unordered in our target workloads, the original input-file order is not preserved across cells; order-sensitive workloads would require an additional global-index stream during decompression.

On a 3GB HACC subset compressed with 4 ranks at $eb = 10^{-3}$, the gather-and-expand decompressor recovers every point within the prescribed bound. On a full 12.9 GB HACC file with 1.07×10^9 points, it sustains above 80% iso efficiency from 1 to 16 nodes for compute and encoding phases, while the shuffle accounts for less than 4% of the total wall-clock time. The distributed implementation retains 99.7% of the monolithic compression ratio at 4 ranks and 99.5% at 16 ranks on both subset and fullset of HACC.

5 EVALUATION

We evaluate XnYZip on 8 point cloud datasets against 7 state-of-the-art baselines. The experiments cover four aspects: theory-practice consistency, end-to-end performance, throughput, and case studies. The datasets include scientific and non-scientific domains, with both dense simulation data and conventional point clouds.

5.1 Experiment Setup

Datasets. We evaluate our framework on eight datasets from diverse scientific and industrial domains, listed in Table 4. Our evaluation focuses on the 3D spatial coordinates (x, y, z) for all datasets. For the original terabyte-scale datasets (EXAALT, USGS, HACC, WarpX) and other large-scale data, we extract representative subsets for evaluation. Smaller datasets, such as Stanford Dragon and GroEL-GroES (1AON), are compressed in their entirety. For datasets with a temporal dimension, such as YIIP and WarpX, we select one representative frame because our work focuses on compressing individual point-cloud snapshots.

Hardware & Software. All experiments are conducted on Bebop Cluster of ANL. Each compute node has dual-socket Intel Xeon E5-2695 v4 processors with 36 physical cores in total and 128 GB DDR4 memory. Our code is compiled using GCC 13.2.0 with the -O3 optimization flag.

Baselines. We compare XnYZip with seven compressors covering scientific error-bounded compression, point cloud compression, and database-side floating-point codecs. Table 5 summarizes their target domains and how they are used in our evaluation.

Table 5: Baseline compressors used in the evaluation.

Baseline	Type	Target Domain
SZ3 [43]	Error-bounded lossy	Scientific arrays
ZFP [45]	Error-bounded lossy	Scientific arrays
LCP [73]	Error-bounded lossy	Scientific particles
Draco [22]	Lossy point cloud	Graphics / media
TMC13 [52]	Lossy point cloud	Graphics / media
ALP [2]	Lossless float codec	Numeric columns
BUFF [46]	Bounded-precision float codec	Numeric columns

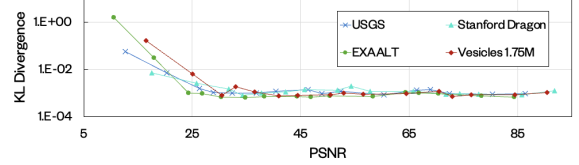


Figure 6: Kullback-Leibler divergence measuring deviation between empirical error distributions and the uniform spatial error assumption across PSNR levels.

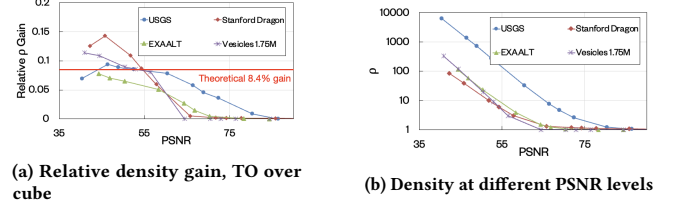


Figure 7: Geometric efficiency of Truncated Octahedron (TO) quantization across different density regimes and datasets.

Table 6: Information-theoretic analysis of different serialization methods on the USGS dataset at $eb = 1$ (PSNR = 66 dB). Hilbert attains the lowest post-delta residual entropy $H(R)$.

Serialization Method	Pre-Delta	Post-Delta	
	Cond. Entropy $H(S_i S_{i-1})$	Mutual Info $I(R_i; R_{i-1})$	Entropy $H(R)$
Row major order	0.5314	0.0082	0.8719
Z-order Curve	0.5364	0.0070	0.8530
Hilbert Curve	0.5324	0.0067	0.8351

5.2 Theory-Practice Consistency Analysis

We validate three theoretical claims of Sections 3–4. (i) The spatial-uniformity assumption underlying our quantization analysis predicts that the empirical per-voxel error distribution matches the one induced by uniformly distributed points within each voxel; Figure 6 measures the gap as a KL divergence, which converges to $O(10^{-3})$ on every dataset as the bound tightens. (ii) The 8.4% theoretical voxel-count advantage of TO over the cube in the high-density regime (Table 3) is recovered empirically in Figure 7: a stable $\approx 8.6\%$ density reduction in the $\rho \gg 1$ regime, smoothly diminishing as $\rho \rightarrow 1$. (iii) The information-theoretic analysis on USGS (Table 6) confirms that the Hilbert curve attains the lowest post-delta residual entropy across the row-major, Z-order, and Hilbert candidates, validating our SFC family.

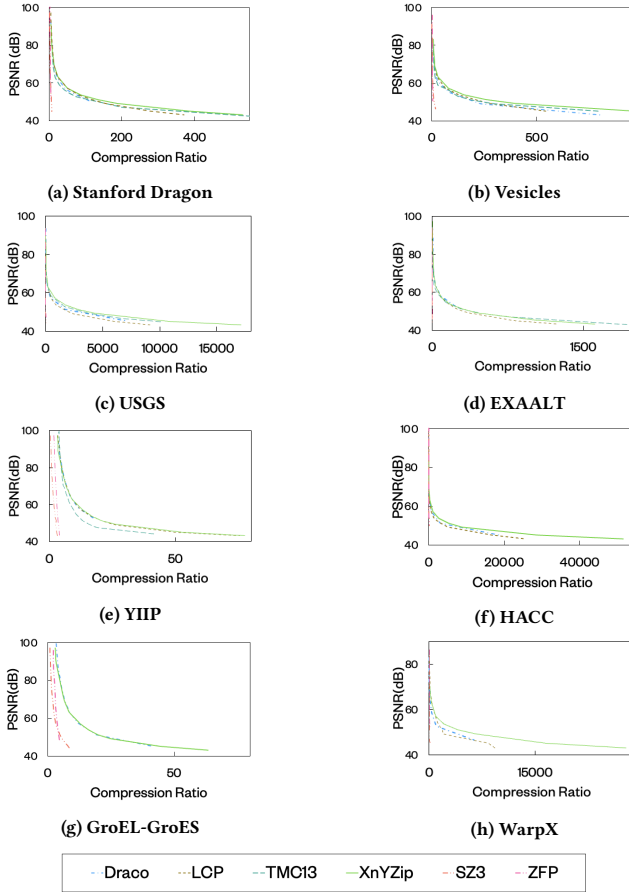


Figure 8: Rate–distortion. XnYZip is at or above the Pareto frontier on every dataset and dominates in the low-PSNR (dense) regime.

5.3 Performance Analysis

Next, we evaluate XnYZip’s end-to-end behaviour against the baselines on the same eight datasets in different metrics.

Rate-distortion. Figure 8 shows the distortion rate per-dataset in all eight evaluation datasets. XnYZip lies at or above the Pareto frontier on every dataset and dominates the low-PSNR (dense) regime predicted by the theory in Section 4; the advantage diminishes smoothly as occupancy thins, exactly as the ρ model.

Besides PSNR, another key distortion metric is the user-specified L2 error bound. Table 7 reports the compression ratios of eight methods on eight datasets at six relative-L2 maximum-error targets, where the L2 error is normalized by the bounding-box diagonal. Each cell gives the highest compression ratio directly measured while satisfying $|p - p'|_2 / \text{diag} \leq \text{target}$. Thus, entries in the same column share the same physical error bound and are directly comparable. A dash indicates that even the method’s tightest measured configuration exceeds the target. BUFF is evaluated at the same L2 targets as a bounded-precision lossy baseline; because its valid bit-precision range is narrow, its ratios remain near $1\times$ and far

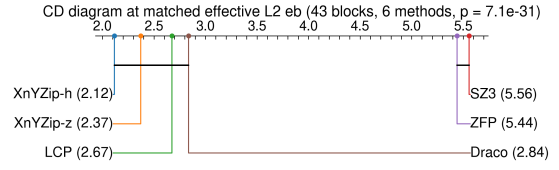


Figure 9: Critical-difference diagram for the six lossy compressors evaluated at matched effective L2 max-error across 43 (dataset, eb) blocks spanning 8 datasets. The Friedman test yields $p = 7.1 \times 10^{-31}$. Methods connected by a horizontal bar are not statistically distinguishable at $\alpha = 0.05$ under the Nemenyi post-hoc test. TMC13 is excluded because its leaves too many invalid entries across the matched-error blocks; BUFF is excluded because its valid bit-precision range is too narrow to populate sufficient blocks.

below those of dedicated geometric codecs. Since ALP is lossless, Table 7 reports one compression ratio per dataset.

To assess the effect of scale, we apply strided subsampling to two real-world datasets, retaining every k -th point for $k \in 32, 16, 8, 4, 2, 1$ to reduce density while preserving spatial extent. Compression ratio increases monotonically with scale at every error bound, showing that subsampling does not weaken our conclusion. For HACC-small, it rises from $23.0\times$ to $93.3\times$ at $eb = 0.2$, while for USGS it rises from $37.4\times$ to $382.3\times$ at $eb = 2$. The same trend holds at all other tested bounds, and all 72 configurations show monotone non-decreasing compression ratios as scale increases.

Statistical significance. A Friedman omnibus test followed by a Nemenyi post-hoc analysis on the seven lossy compressors at various L2 max-error (Figure 9) places XnYZip-h/-z, Draco, and LCP in a top cluster statistically separated from ZFP and SZ3.

Visual quality. Beyond aggregate statistics, Figure 10 compares reconstructions of the Stanford Dragon at a fixed compression ratio of 110. The isotropic BCC lattice yields smoother surfaces and avoids the streak-like artefacts visible in the LCP and Draco reconstructions at a comparable PSNR.

SFC choice. Table 8 compares the compression ratios of the Hilbert and Z-order variants across all datasets. The gap remains within 6.4% on seven of the eight datasets; WarpX is an outlier where the highly clustered plasma coordinates make Z-order’s cheaper mapping a better match for the subsequent delta+RLE stages. Overall, we recommend XnYZip-z as the default and XnYZip-h as the compression-oriented alternative for typical workloads.

Database-side baselines. We additionally evaluate two database-oriented floating-point codecs: ALP (lossless) and BUFF (bit-precision-bounded, using the authors’ Tranway1/buff implementation). ALP provides modest lossless gains on the four datasets where its build completes successfully, while BUFF achieves only low single-digit per-axis compression ratios within its supported bit-precision range. Both remain substantially below the lossy geometric codecs at every matched-L2 operating point in Table 7, highlighting the different rate-distortion characteristics of one-dimensional column codecs and three-dimensional point-cloud compressors.

Metadata overhead at extreme CR. At very high CR, the bitstream becomes sufficiently small that the per-block Huffman dictionary dominates the total size: the long tail of low-frequency symbols still requires explicit representation, and the dictionary

Table 7: Master compression-ratio table under the standard error-bounded rate-distortion convention. Each cell reports the best measured CR that satisfies the corresponding relative-L2 max-error target. A dash means the method’s tightest measured configuration still exceeds the target or met bugs. The ALP row reports the lossless ALP result for reference.

Error	YIIP						Stanford Dragon						GroEL-GroES						Vesicles					
	1e-5	5e-5	1e-4	5e-4	1e-3	5e-3	1e-5	5e-5	1e-4	5e-4	1e-3	5e-3	1e-5	5e-5	1e-4	5e-4	1e-3	5e-3	1e-5	5e-5	1e-4	5e-4	1e-3	5e-3
SZ3	0.62	0.62	0.78	1.30	1.60	2.40	0.97	1.36	1.62	2.55	3.26	6.02	0.68	1.03	1.21	1.83	2.28	4.78	1.09	1.45	1.68	2.67	3.56	10.58
ZFP	1.54	1.80	1.91	2.17	2.33	2.72	1.79	2.13	2.27	2.63	2.86	3.44	1.69	2.01	2.14	2.47	2.68	3.22	1.77	2.13	2.28	2.66	2.90	3.53
LCP	3.05	3.92	4.47	6.59	8.30	21.62	4.19	5.67	6.61	12.99	20.17	124.3	3.02	3.85	4.37	6.50	8.17	20.45	3.90	5.48	6.61	12.23	21.40	219.2
Draco	2.71	3.64	4.11	5.52	6.68	11.47	3.49	5.18	6.16	9.82	13.80	46.74	3.00	4.17	4.79	6.84	8.71	19.21	3.90	6.13	7.60	14.63	28.14	231.6
TMC13	–	–	–	–	3.81	3.81	–	–	–	–	–	–	–	–	4.47	6.24	8.47	18.20	3.63	5.55	5.85	13.19	29.08	189.5
BUFF	1.68	2.13	2.13	2.59	2.59	2.59	1.71	1.71	2.18	2.59	2.59	3.84	1.88	2.00	2.00	2.00	2.00	2.00	1.71	1.71	1.71	1.71	1.71	1.71
ALP	2.34 (lossless)						1.32 (lossless)						1.68 (lossless)						1.14 (lossless)					
XnYZip-h	2.81	3.43	4.33	6.79	9.00	26.15	3.29	5.51	6.69	14.20	23.76	187.6	2.74	3.31	4.20	6.56	8.55	24.78	3.55	5.63	6.85	15.09	28.73	391.1
XnYZip-z	2.82	3.43	4.33	6.78	8.96	25.86	3.29	5.51	6.69	13.92	22.41	180.2	2.74	3.31	4.21	6.45	8.65	24.17	3.55	5.63	6.80	14.73	27.84	373.6

Error	USGS						EXAALT						HACC						WarpX					
	1e-5	5e-5	1e-4	5e-4	1e-3	5e-3	1e-5	5e-5	1e-4	5e-4	1e-3	5e-3	1e-5	5e-5	1e-4	5e-4	1e-3	5e-3	1e-5	5e-5	1e-4	5e-4	1e-3	5e-3
SZ3	1.25	1.52	1.78	2.80	3.75	7.65	0.75	0.92	1.02	1.31	1.49	2.23	1.46	2.13	2.66	5.87	9.81	84.34	2.52	3.30	3.87	5.58	6.89	14.94
ZFP	1.84	2.08	2.22	2.57	2.79	3.66	1.63	1.82	1.93	2.36	2.54	3.03	1.97	2.25	2.42	2.85	3.12	4.19	2.09	2.48	2.63	2.99	3.21	3.76
LCP	5.40	9.14	13.05	54.31	137.6	1685.0	4.06	5.78	7.06	15.17	27.53	265.7	–	11.02	16.10	62.50	160.8	3406.0	10.86	22.84	30.83	111.5	272.4	1956.9
Draco	–	7.85	10.40	27.15	59.68	436.7	3.58	5.40	6.50	10.94	16.61	85.38	6.37	14.63	22.28	76.97	209.9	2968.2	4.06	6.35	7.82	32.95	88.39	968.37
TMC13	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BUFF	1.75	1.75	1.75	1.75	1.75	1.75	1.88	2.13	2.13	2.13	2.13	2.13	1.85	2.09	2.09	2.09	2.09	2.09	1.60	2.13	2.29	2.29	2.29	3.56
ALP	1.40 (lossless)						1.18 (lossless)						1.15 (lossless)						1.22 (lossless)					
XnYZip-h	5.44	6.63	14.55	69.78	210.6	4080.0	3.81	5.87	7.27	17.68	36.25	434.6	6.25	12.57	19.15	84.90	248.6	8778.6	5.22	7.63	10.66	80.70	327	6603
XnYZip-z	5.46	6.65	14.48	68.38	207.9	4112.3	3.81	5.87	7.28	17.67	36.03	429.1	6.22	12.45	18.95	84.03	247.4	8894.2	5.70	11.94	13.26	102.0	336	6662

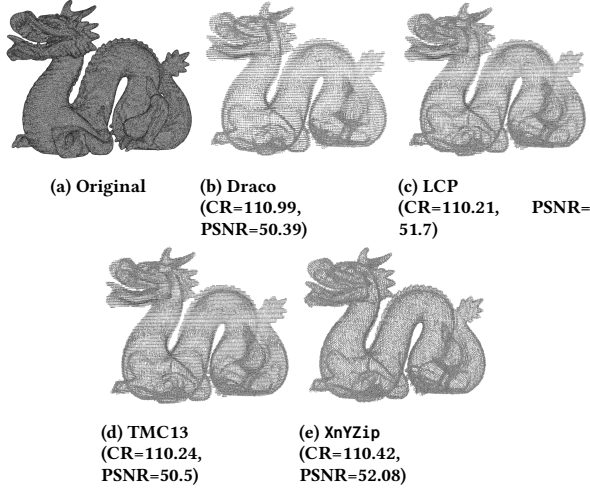


Figure 10: Stanford Dragon reconstructions at a fixed CR of 110. XnYZip preserves surface smoothness without the streak artefacts of LCP/Draco.

alone can account for more than 60% of the bitstream, limiting the achievable CR. Replacing the metadata stream with an Exp-Golomb code (as used by TMC13 for this purpose) would avoid explicit dictionary materialization and alleviate this limitation without changing the underlying quantization geometry. We view this as an implementation-level optimization orthogonal to the main contributions of the paper.

5.4 Throughput

We evaluate throughput in two steps: (1) the constant-factor overhead of TO over cube quantization, and (2) end-to-end compression and decompression throughput.

Table 8: Compression ratio of XnYZip-h (Hilbert) and XnYZip-z (Z-order) on every dataset at $eb = 10^{-3}$, and the worst-case CR gap observed across the per-dataset 19-eb sweep. Positive gap values indicate Hilbert wins, negative values indicate Z-order wins.

Dataset	CR (-h)	CR (-z)	gap @ $eb \approx 10^{-3}$	worst-case gap
YIIP	9.00	8.96	+0.4%	+2.8%
Stanford Dragon	23.76	22.41	+6.0%	+6.0%
GroEL-GroES (1AON)	8.55	8.65	–1.1%	+3.4%
Vesicles	28.73	27.84	+3.2%	+6.4%
USGS	210.65	207.94	+1.3%	+2.1%
EXAALT	36.25	36.03	+0.6%	+2.4%
HACC	248.61	247.36	+0.5%	–2.2%
WarpX	266.52	277.88	–4.1%	–37.4%

Table 9: Cube vs. TO quantization timing inside the production compressor ($eb = 10^{-3}$, single-threaded). *Quantize* is quantization stage time; *Total* is end-to-end compression time. Each cell is the average of three runs.

Dataset	Quantize (ms)		Total (s)		TO ratio TO/cube total
	cube	TO	cube	TO	
Stanford Dragon	3.1	7.9	0.036	0.038	1.06×
Vesicles	13.7	38.8	0.305	0.355	1.16×
USGS	99.8	340.4	3.39	3.61	1.07×
HACC (22.4 M)	133.3	509.3	3.89	4.00	1.03×

TO vs cube constant factor. We measure quantization inside the production binary with chrono timers and run the binary with the cube and the TO quantizer on the same input. Within the pipeline the TO call costs higher (Table 9) but the quantization stage is only 3–13% of total compression time, the end-to-end overhead of TO vs cube is just +3–+16% wall-clock, while delivering a substantially higher CR on every dataset tested. The two-candidate parity check in TO quantization explains this constant, which is bounded and independent of N .

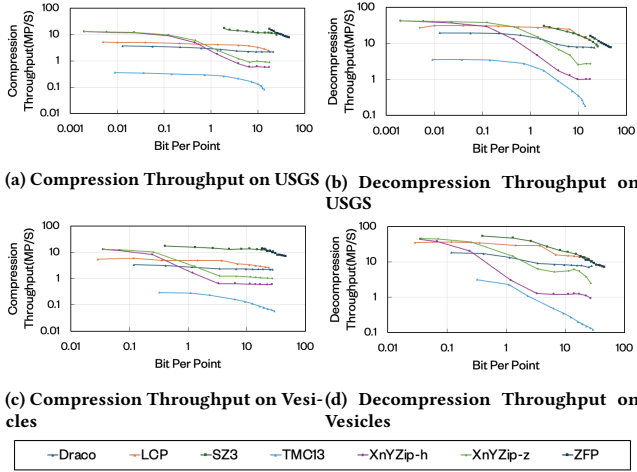


Figure 11: Throughput vs. bitrate on USGS and Vesicles. At low bpps (dense regime, RLE active) XnYZip attains the highest throughput; at higher bpps (RLE off) it remains competitive with Draco/LCP and ahead of TMC13.

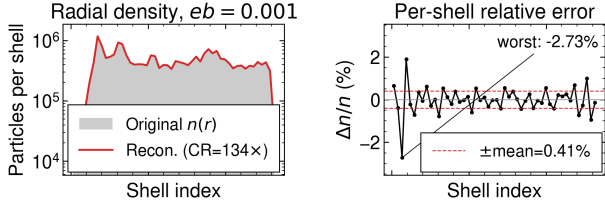


Figure 12: HACC radial density profile at $eb = 10^{-3}$ ($CR \approx 134\times$). Left: original particle counts per spherical shell and the reconstructed profile coincide to within line thickness across all 50 shells. Right: per-shell relative error $\Delta n/n$. The mean absolute deviation is 0.41%; the worst single shell sits at -2.73% on a 22.4,M-point subset.

End-to-end throughput. Figure 11 plots throughput against bits-per-point on USGS and Vesicles against every baseline. The mapping and entropy coding pipeline (Huffman + Zstd) dominates the wall clock, not the quantization step. At lower bpps, where TO quantization saturates RLE, our framework delivers the highest throughput of any dedicated point-cloud compressor evaluated; at higher bpps, where RLE is disabled, it remains competitive with Draco and LCP and consistently above TMC13.

5.5 Querying the Compressed Representation

Compression is useful only if downstream tasks remain accurate and, ideally, operate directly on compressed data. The chunked representation in Section 4.4 supports both: each chunk’s RLE cell table forms a spatial histogram of cell counts, while a 28-byte sidecar storing the bounding box and total count allows region queries to skip disjoint chunks. Radial density profiles, halo-finder occupancy maps, and radius/range filters therefore reduce to regional count queries. As a representative case, the cosmological radial density profile $n(r)$ partitions space into N_{bin} concentric shells and counts

Table 10: Per-query radius-count latency on Stanford Dragon ($eb = 10^{-3}$), 50 random ball centers. FULL = decode + expand chunks, brute-force filter. INDEX = the bbox-skip + cell-count path. “skip” is the fraction of (chunk, query) pairs for which the sidecar bbox is disjoint from the ball; INDEX counts match FULL bit-exactly on every row.

chunks	r/diag	FULL (ms)	INDEX (ms)	speedup	skip
8	0.02	0.179	0.010	$17.1\times$	66%
8	0.05	0.176	0.017	$10.6\times$	52%
8	0.10	0.181	0.024	$7.5\times$	38%
32	0.02	0.195	0.006	$34.6\times$	84%
32	0.05	0.183	0.009	$19.9\times$	74%
32	0.10	0.181	0.016	$11.4\times$	61%

particles in each shell, providing a standard input to halo, void, and overdensity analyses. We compute $n(r)$ using 50 shells on a 22.4,M-point HACC subset and its reconstruction at $CR \approx 134\times$ and $eb = 10^{-3}$. The reconstructed profile differs from the original by at most 2.73% per shell, with a mean deviation of 0.41% (Figure 12). The largest errors occur in the innermost shells, where smaller volumes produce lower and noisier particle counts.

The same query runs *without* full decompression: test each chunk’s bbox against the query region, use the sidecar for fully contained chunks, and decode only boundary chunks. On Stanford Dragon at $eb = 10^{-3}$ over 50 random ball centers, this bit-exact path is 7.5–34.6 $\times$ faster than decode-then-filter (Table 10); the 28-B sidecar costs $\leq 0.05\%$ of the compressed bytes on realistic chunk counts and at most 1.43% in a stressed 32-chunk Dragon configuration. The same compressed bytes therefore serve as storage codec, spatial-histogram index, and per-region aggregator.

6 CONCLUSION AND FUTURE WORK

In this paper, we identify a fundamental mismatch in lossy point cloud compression: widely used L2-based distortion metrics and suboptimal cubic quantization. We formulate the problem as optimal space-filling lattice selection and show that, under MSE distortion, the BCC lattice with Truncated Octahedron Voronoi cells provides the optimal geometry. Based on this result, we develop XnYZip, which integrates TO-based quantization with a locality-aware SFC/RLE encoding pipeline. Compared with state-of-the-art compressors, XnYZip achieves up to 3 $\times$ higher compression ratios, 2.2 $\times$ and 1.2 $\times$ faster compression and decompression throughput, respectively, on dense scientific point cloud datasets. Future work will explore adaptive hybrid quantization for sparse regions, multi-frame point clouds, and parallel CPU/GPU implementations for high-throughput *in-situ* compression.

7 ACKNOWLEDGEMENT

The material was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR), under contract DE-AC02-06CH11357, and supported by the U.S. National Science Foundation under Grant OAC-2311875, OAC-2344717, OAC-2513768, OAC-2514036, OAC-2514034, OAC-2609480, OAC-2542646, and III-2107213. We acknowledge the computing resources provided on Bebop (operated by Laboratory Computing Resource Center at Argonne).

REFERENCES

- [1] 2023. *HDF5 Filters*. https://docs.hdfgroup.org/hdf5/develop/_f_i_l_t_e_r.html Online.
- [2] Azim Afroozeh, Leonardo X. Kuffo, and Peter Boncz. 2023. ALP: Adaptive Lossless floating-Point Compression. *Proc. ACM Manag. Data* 1, 4, Article 230 (Dec. 2023), 26 pages. <https://doi.org/10.1145/3626717>
- [3] Nitin Agrawal and Ashish Vulimiri. 2017. Low-Latency Analytics on Colossal Data Streams with SummaryStore. In *Proceedings of the 26th Symposium on Operating Systems Principles* (Shanghai, China) (SOSP '17). Association for Computing Machinery, New York, NY, USA, 647–664. <https://doi.org/10.1145/3132747.3132758>
- [4] Mark Ainsworth, Ozan Tugluk, Ben Whitney, and Scott Klasky. 2018. Multilevel techniques for compression and reduction of scientific data—the univariate case. *Computing and Visualization in Science* 19, 5–6 (2018), 65–76.
- [5] Chao Cao, Marius Preda, and Titus Zaharia. 2019. 3D Point Cloud Compression: A Survey. In *Proceedings of the 24th International Conference on 3D Web Technology* (LA, CA, USA) (Web3D '19). Association for Computing Machinery, New York, NY, USA, 1–9. <https://doi.org/10.1145/3329714.3338130>
- [6] Yufan Chen, Xiangyu Zou, Kaiwen Deng, Hao Hu, Cai Deng, Ke Feng, and Wen Xia. 2025. Apic: A Precomputation-Based Integer Compressor for OLTP Databases. In *2025 Data Compression Conference (DCC)*. 303–312. <https://doi.org/10.1109/DCC62719.2025.00038>
- [7] Zhiyuan Chen, Johannes Gehrke, and Flip Korn. 2001. Query optimization in compressed database systems. 271–282.
- [8] Yu Cheng and Florin Rusu. 2014. Parallel in-situ data processing with speculative loading. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data* (Snowbird, Utah, USA) (SIGMOD '14). Association for Computing Machinery, New York, NY, USA, 1287–1298. <https://doi.org/10.1145/2588555.2593673>
- [9] Community Earth System Model (CESM) Atmosphere Model. 2019. <http://www.cesm.ucar.edu/models/>. Online.
- [10] J. Conway and N. Sloane. 1982. Voronoi regions of lattices, second moments of polytopes, and quantization. *IEEE Transactions on Information Theory* 28, 2 (1982), 211–226. <https://doi.org/10.1109/TIT.1982.1056483>
- [11] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory* (Wiley Series in Telecommunications and Signal Processing). Wiley-Interscience, USA.
- [12] cuZFP. 2023. https://github.com/LLNL/zfp/tree/develop/src/cuda_zfp. Online.
- [13] Ismael Daribo, Ryo Furukawa, Ryusuke Sagawa, Hiroshi Kawasaki, Shinsaku Hiura, and Naoki Asada. 2012. Efficient rate-distortion compression of dynamic point cloud for grid-pattern-based 3D scanning systems. *3D Research* 3, 1 (17 Jan 2012), 2. [https://doi.org/10.1007/3DRes.01\(2012\)2](https://doi.org/10.1007/3DRes.01(2012)2)
- [14] Michael Deering. 1995. Geometry compression. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (SIGGRAPH '95). Association for Computing Machinery, New York, NY, USA, 13–20. <https://doi.org/10.1145/218380.218391>
- [15] Sheng Di. 2023. *H5Z-SZ*. <https://github.com/disheng222/H5Z-SZ> Online.
- [16] Sheng Di and Franck Cappello. 2016. Fast error-bounded lossy HPC data compression with SZ. In *2016 IEEE International Parallel and Distributed Processing Symposium*. IEEE, IEEE, Chicago, IL, USA, 730–739.
- [17] EXAALT Project. 2021. EXAALT: Molecular Dynamics at Exascale for Materials Science. <https://www.exascaleproject.org/research-project/exaalt/>. Online.
- [18] Haoqiang Fan, Hao Su, and Leonidas Guibas. 2017. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2463–2471. <https://doi.org/10.1109/CVPR.2017.264>
- [19] L. Fedeli, A. Huebl, F. Boillot-Cerneux, T. Clark, K. Gott, C. Hillairet, S. Jaure, A. Leblanc, R. Lehe, A. Myers, C. Piechurski, M. Sato, N. Zaim, W. Zhang, J. Vay, and H. Vincenti. 2022. Pushing the Frontier in the Design of Laser-Based Electron Accelerators with Groundbreaking Mesh-Refined Particle-In-Cell Simulations on Exascale-Class Supercomputers. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Computer Society, Los Alamitos, CA, USA, 1–12. <https://doi.org/10.1109/SC41404.2022.00008>
- [20] Jianyang Gao and Cheng Long. 2024. RaBitQ: Quantizing High-Dimensional Vectors with a Theoretical Error Bound for Approximate Nearest Neighbor Search. *Proc. ACM Manag. Data* 2, 3, Article 167 (May 2024), 27 pages. <https://doi.org/10.1145/3654970>
- [21] A. Gersho. 1979. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory* 25, 4 (1979), 373–380. <https://doi.org/10.1109/TIT.1979.1056067>
- [22] Google. 2017. Draco: A library for compressing and decompressing 3D geometric meshes and point clouds. <https://github.com/google/draco>. Online.
- [23] Jim Gray, David T. Liu, Maria Nieto-Santisteban, Alex Szalay, David J. DeWitt, and Gerd Heber. 2005. Scientific data management in the coming decade. *SIGMOD Rec.* 34, 4 (Dec. 2005), 34–41. <https://doi.org/10.1145/1107499.1107503>
- [24] Stefan Gumhold, Zachi Kami, Martin Isenbaur, and Hans-Peter Seidel. 2005. Predictive point-cloud compression. In *ACM SIGGRAPH 2005 Sketches* (Los Angeles, California) (SIGGRAPH '05). Association for Computing Machinery, New York, NY, USA, 137–es. <https://doi.org/10.1145/1187112.1187277>
- [25] HACC team (ECP EXASKY). 2019. <https://sdrbench.github.io/>. Online.
- [26] David Hilbert. 1891. Ueber die stetige Abbildung einer Linie auf ein Flächenstück. *Math. Ann.* 38, 3 (01 Sep 1891), 459–460. <https://doi.org/10.1007/BF01199431>
- [27] Kaisei Hishida, Chunwei Liu, John Paparrizos, and Aaron J. Elmore. 2025. Beyond Compression: A Comprehensive Evaluation of Lossless Floating-Point Compression. *Proc. VLDB Endow.* 18, 11 (July 2025), 4396–4409. <https://doi.org/10.14778/3749646.3749701>
- [28] Hao Hu, Qiyang Zheng, Xiangyu Zou, Lisha Qin, Chengwei Zhang, Wanchuan Zhang, Zhaohe Jiang, Dingwen Tao, Hongpeng Wang, and Wen Xia. 2025. A Cost-Effective and Decompression-Transparent Compressor for OLTP-Oriented Databases. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. 405–418. <https://doi.org/10.1109/ICDE65448.2025.00037>
- [29] Yan Huang, Jingliang Peng, C.-C. Jay Kuo, and M. Gopi. 2006. Octree-Based Progressive Geometry Coding of Point Clouds. In *Symposium on Point-Based Graphics*, Mario Botsch, Baoquan Chen, Mark Pauly, and Matthias Zwicker (Eds.). The Eurographics Association. <https://doi.org/10.2312/SPBG/SPBG06/103-110>
- [30] Alibay Irfan, Beckstein Oliver, Fan Shujie, J. Gowers Richard, Matta Micaela, and Wang Lily. 2018. YiiP equilibrium dataset. https://www.mdanalysis.org/MDAnalysisData/yiiP_equilibrium.html. Online.
- [31] Balakrishna R. Iyer and David Wilhite. 1994. Data Compression Support in Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 695–704.
- [32] Søren Keiser Jensen, Torben Bach Pedersen, and Christian Thomsen. 2018. ModelarDB: modular model-based time series management with spark and cassandra. *Proc. VLDB Endow.* 11, 11 (July 2018), 1688–1701. <https://doi.org/10.14778/3236187.3236215>
- [33] Pu Jiao, Sheng Di, Hanqi Guo, Kai Zhao, Jiannan Tian, Dingwen Tao, Xin Liang, and Franck Cappello. 2022. Toward Quantity-of-Interest Preserving Lossy Compression for Scientific Data. *Proc. VLDB Endow.* 16, 4 (Dec. 2022), 697–710. <https://doi.org/10.14778/3574245.3574255>
- [34] Sian Jin, Sheng Di, Jiannan Tian, Suren Byna, Dingwen Tao, and Franck Cappello. 2022. Improving Prediction-Based Lossy Compression Dramatically via Ratio-Quality Modeling. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. 2494–2507. <https://doi.org/10.1109/ICDE53745.2022.00232>
- [35] Joint Photographic Experts Group. 1992. JPEG: Still Image Data Compression Standard (ISO/IEC 10918-1). ITU-T Recommendation T.81 and ISO/IEC 10918-1 Standard. Online; accessed on 2025-10-17.
- [36] Herve Jégou, Matthijs Douze, and Cordelia Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1 (2011), 117–128.
- [37] Julius Kammerl, Nico Blodow, Radu Bogdan Rusu, Suat Gedikli, Michael Beetz, and Eckehard Steinbach. 2012. Real-time compression of point cloud streams. In *2012 IEEE International Conference on Robotics and Automation*. 778–785. <https://doi.org/10.1109/ICRA.2012.6224647>
- [38] Birendra Kathariya, Li Li, Zhu Li, Jose Alvarez, and Jianle Chen. 2018. Scalable Point Cloud Geometry Coding with Binary Tree Embedded Quadtree. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*. 1–6. <https://doi.org/10.1109/ICME.2018.8486481>
- [39] Ian M. Kenney and Oliver Beckstein. 2015. Technical Report: SPIDAL Summer REU 2015: Biomolecular benchmark systems. (10 2015). <https://doi.org/10.6084/m9.figshare.1588804.v1>
- [40] Martin L. Kersten, Stratos Idreos, Stefan Manegold, and Erietta Liarou. 2011. The researcher's guide to the data deluge: querying a scientific database in just a few seconds. *Proc. VLDB Endow.* 4, 12 (Aug. 2011), 1474–1477. <https://doi.org/10.14778/3402755.3402799>
- [41] Masaki Kitago and M. Gopi. 2006. Efficient and Prioritized Point Subsampling for CSRBF Compression. In *Symposium on Point-Based Graphics*, Mario Botsch, Baoquan Chen, Mark Pauly, and Matthias Zwicker (Eds.). The Eurographics Association. <https://doi.org/10.2312/SPBG/SPBG06/121-128>
- [42] Xin Liang, Sheng Di, Dingwen Tao, Sihuan Li, Shaomeng Li, Hanqi Guo, Zizhong Chen, and Franck Cappello. 2018. Error-controlled lossy compression optimized for high compression ratios of scientific datasets. In *2018 IEEE International Conference on Big Data*. IEEE, 438–447.
- [43] Xin Liang, Kai Zhao, Sheng Di, Sihuan Li, Robert Underwood, Ali M. Gok, Jiannan Tian, Junjing Deng, Jon C. Calhoun, Dingwen Tao, Zizhong Chen, and Franck Cappello. 2022. SZ3: A Modular Framework for Composing Prediction-Based Error-Bounded Lossy Compressors. *IEEE Transactions on Big Data* (2022), 1–14. <https://doi.org/10.1109/TBDDATA.2022.3201176>
- [44] Jyh-Ming Lien, Gregorij Kurillo, and Ruzena Bajcsy. 2010. Multi-camera tele-immersion system with real-time model driven data compression. *The Visual Computer* 26, 1 (01 Jan 2010), 3–15. <https://doi.org/10.1007/s00371-009-0367-8>
- [45] Peter Lindstrom. 2014. Fixed-rate compressed floating-point arrays. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2674–2683.
- [46] Chunwei Liu, Hao Jiang, John Paparrizos, and Aaron J. Elmore. 2021. Decomposed bounded floats for fast compression and queries. *Proc. VLDB Endow.* 14, 11 (July 2021), 2586–2598. <https://doi.org/10.14778/3476249.3476305>

- [47] Jinyang Liu, Pu Jiao, Kai Zhao, Xin Liang, Sheng Di, and Franck Cappello. 2025. QPET: A Versatile and Portable Quantity-of-Interest-Preservation Framework for Error-Bounded Lossy Compression. *Proc. VLDB Endow.* 18, 8 (April 2025), 2440–2453. <https://doi.org/10.14778/3742728.3742739>
- [48] T.D. Lookabaugh and R.M. Gray. 1989. High-resolution quantization theory and the vector quantizer advantage. *IEEE Transactions on Information Theory* 35, 5 (1989), 1020–1033. <https://doi.org/10.1109/18.42217>
- [49] Philipp Merkle, Aljoscha Smolic, Karsten Muller, and Thomas Wiegand. 2007. Multi-View Video Plus Depth Representation and Coding. In *2007 IEEE International Conference on Image Processing*, Vol. 1. 1 – 201–I – 204. <https://doi.org/10.1109/ICIP.2007.4378926>
- [50] G. M. Morton. 1966. *A Computer Oriented Geodetic Data Base; and a New Technique in File Sequencing*. Technical Report. IBM Ltd., Ottawa, Canada. Technical Report.
- [51] MPEG. 2024. MPEG-I Part 9: Geometry-based Point Cloud Compression. <https://www.mpeg.org/standards/MPEG-I/9/>. Accessed: 2024-03-22.
- [52] MPEG Group. 2017. Geometry based point cloud compression (G-PCC) test model. <https://github.com/MPEGGroup/mpeg-pcc-tmc13>. Online.
- [53] NYX simulation. 2019. <https://amrex-astro.github.io/Nyx/>. Online.
- [54] Tilo Ochotta and Dietmar Saupe. 2004. Compression of point-based 3D models by shape-adaptive wavelet coding of multi-height fields. In *Proceedings of the First Eurographics Conference on Point-Based Graphics (Switzerland) (SPBG'04)*. Eurographics Association, Goslar, DEU, 103–112.
- [55] Meikel Poess and Dmitry Potapov. 2003. Data compression in Oracle. In *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29 (Berlin, Germany) (VLDB '03)*. VLDB Endowment, 937–947.
- [56] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A. Chou, Robert A. Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, Joan Llach, Khaled Mammou, Rafael Mekuria, Ohji Nakagami, Ernestasia Siahaan, Ali Tabatabai, Alexis M. Tourapis, and Vladyslav Zakharchenko. 2019. Emerging MPEG Standards for Point Cloud Compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (2019), 133–148. <https://doi.org/10.1109/JETCAS.2018.2885981>
- [57] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A. Chou, Robert A. Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, Joan Llach, Khaled Mammou, Rafael Mekuria, Ohji Nakagami, Ernestasia Siahaan, Ali Tabatabai, Alexis M. Tourapis, and Vladyslav Zakharchenko. 2019. Emerging MPEG Standards for Point Cloud Compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (2019), 133–148. <https://doi.org/10.1109/JETCAS.2018.2885981>
- [58] Stanford University Computer Graphics Laboratory. 1996. The Stanford 3D Scanning Repository. <https://graphics.stanford.edu/data/3Dscanrep/>. Online.
- [59] Dingwen Tao, Sheng Di, Zizhong Chen, and Franck Cappello. 2017. Significantly improving lossy compression for scientific data sets based on multidimensional prediction and error-controlled quantization. In *2017 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 1129–1139.
- [60] Gabriel Taubin and Jarek Rossignac. 1998. Geometric compression through topological surgery. *ACM Trans. Graph.* 17, 2 (April 1998), 84–115. <https://doi.org/10.1145/274363.274365>
- [61] David Taubman and Michael Marcellin. 2012. *JPEG2000 image compression fundamentals, standards and practice: image compression fundamentals, standards and practice*. Vol. 642. Springer Science & Business Media.
- [62] The HDF Group. 2023. *Hierarchical data format version 5*. <http://www.hdfgroup.org/HDF5> Online.
- [63] Hermann Töpf and H. Herzog. 1981. Multidimensional Range Search in Dynamically Balanced Trees. *Angew. Inform.* 23 (1981), 71–77. <https://api.semanticscholar.org/CorpusID:26857103>
- [64] U.S. Geological Survey. 2024. USGS 3DEP LiDAR Point Cloud Dataset. <https://www.usgs.gov/news/technical-announcement/usgs-3dep-lidar-point-cloud-now-available-amazon-public-dataset/>. Online.
- [65] Gregory K Wallace. 1992. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics* 38, 1 (1992), xviii–xxxiv.
- [66] Daoce Wang, Jesus Pulido, Pascal Grosset, Sian Jin, Jiannan Tian, James Ahrens, and Dingwen Tao. 2022. TAC: Optimizing Error-Bounded Lossy Compression for Three-Dimensional Adaptive Mesh Refinement Simulations. In *Proceedings of the 31st International Symposium on High-Performance Parallel and Distributed Computing*. 135–147.
- [67] Mingze Xia, Sheng Di, Franck Cappello, Pu Jiao, Kai Zhao, Jinyang Liu, Xuan Wu, Xin Liang, and Hanqi Guo. 2024. Preserving Topological Feature with Sign-of-Determinant Predicates in Lossy Compression: A Case Study of Vector Field Critical Points. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 4979–4992. <https://doi.org/10.1109/ICDE60146.2024.00378>
- [68] Mingze Xia, Yuxiao Li, Pu Jiao, Bei Wang, Xin Liang, and Hanqi Guo. 2026. Time-varying Vector Field Compression with Preserved Critical Point Trajectories. *arXiv:2510.25143 [cs.DB]* <https://arxiv.org/abs/2510.25143>
- [69] Mingze Xia, Bei Wang, Yuxiao Li, Pu Jiao, Xin Liang, and Hanqi Guo. 2025. TspSZ: An Efficient Parallel Error-Bounded Lossy Compressor for Topological Skeleton Preservation. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. IEEE Computer Society, Los Alamitos, CA, USA, 3682–3695. <https://doi.org/10.1109/ICDE65448.2025.00275>
- [70] Zhaohui Xu, Arthur L. Horwich, and Paul B. Sigler. 1997. The crystal structure of the asymmetric GroEL–GroES–(ADP)7 chaperonin complex. *Nature* 388, 6644 (01 Aug 1997), 741–750. <https://doi.org/10.1038/41944>
- [71] Feng Zhang, Zaifeng Pan, Yanliang Zhou, Jidong Zhai, Xipeng Shen, Onur Mutlu, and Xiaoyong Du. 2021. G-TADOC: Enabling Efficient GPU-Based Text Analytics without Decompression. In *ICDE*. 1679–1690.
- [72] Feng Zhang, Jidong Zhai, Xipeng Shen, Onur Mutlu, and Wenguang Chen. 2018. Efficient document analytics on compressed data: method, challenges, algorithms, insights. *Proc. VLDB Endow.* 11, 11 (July 2018), 1522–1535.
- [73] Longtao Zhang, Ruoyu Li, Congrong Ren, Sheng Di, Jinyang Liu, Jiajun Huang, Robert Underwood, Pascal Grosset, Dingwen Tao, Xin Liang, Hanqi Guo, Franck Cappello, and Kai Zhao. 2025. LCP: Enhancing Scientific Data Management with Lossy Compression for Particles. *Proc. ACM Manag. Data* 3, 1, Article 50 (Feb. 2025), 27 pages. <https://doi.org/10.1145/3709700>