Neural Compression for Scientific Particle Data: Perspectives on How Learning Can Help

Tian Qiu[1] **Carolina Cuesta-Lazaro**[2,3] **Sheng Di**[4] **Haewon Jeong**[1,2]
[1] UC Santa Barbara [2] Flatiron Institute, Simons Foundation
[3] Institute for Advanced Study [4] Argonne National Laboratory
{tian_qiu,haewon}@ucsb.edu, ccuesta-lazaro@flatironinstitute.org, sdi1@anl.gov

Abstract

As particle-based simulations grow in scale, efficient compression of scientific data becomes critical. Neural point cloud compression (PCC) methods excel on computer vision (CV) benchmarks, but do they transfer to scientific particle data? We highlight that scientific particles differ from CV point clouds in fractal dimension: CV point clouds are locally planar, but particles are not. This key difference limits the effectiveness of octree data structure that underpins most PCC methods. Neural codecs achieve only marginal bitrate reductions over the traditional G-PCC baseline on scientific data, revealing a significant performance gap. Supported by an evaluation spanning domains of cosmology, molecular dynamics, and plasma physics, we outline where the structure of scientific data limits current neural approaches and where learning can help in the future.

1 Introduction

Particle-based simulation has become a fundamental tool in the physical sciences. It enables us to build high-fidelity digital twins of complex systems. The same fidelity, however, generates rapidly growing data volumes, making efficient compression an urgent need. Cosmological N-body simulations [39, 29, 17] now model the interactions of billions of dark matter particles to study the formation and evolution of large-scale structures in the Universe. Molecular dynamics (MD) simulations [1, 23, 21] resolve atomic-scale phenomena, such as metal deposition [9], protein folding [27], and drug interactions [10], generating long trajectories of millions of atoms. As computational capabilities and simulation fidelity continue to scale, the resulting datasets routinely reach terabyte scale [6], imposing substantial burdens on storage, data movement, and downstream analysis. Consequently, efficient compression of scientific particle data has become a pressing and shared need across disciplines.

Compression of scientific data has received growing attention in high-performance computing community, with developments in both structured grid data and unstructured particle data. Compressors for structured grids [26, 28] leverage strong local correlation between storage-adjacent elements in various ways to achieve compression. However, particle data poses a unique challenge: spatial location is part of the data itself, and storage order often breaks spatial correlation. Current methods for particle data [42] typically operate on individual coordinates or local neighbourhoods, and do not exploit the large-scale spatial structure present in the data, leaving significant room for improvement.

Recent advances in neural point cloud compression (PCC) in the computer vision community raise a compelling opportunity to significantly improve compression for scientific particle data. Traditional PCC methods are primarily designed for computer vision workloads such as LiDAR scans and human body captures. The dominant standard, MPEG G-PCC [31], represents point clouds using an octree structure and relies on hand-crafted context models to entropy-code occupancy information. In contrast, neural PCC approaches [16] learn data-driven predictive models, consistently achieving

Conference on Physics and AI at Stanford University (PAI26).

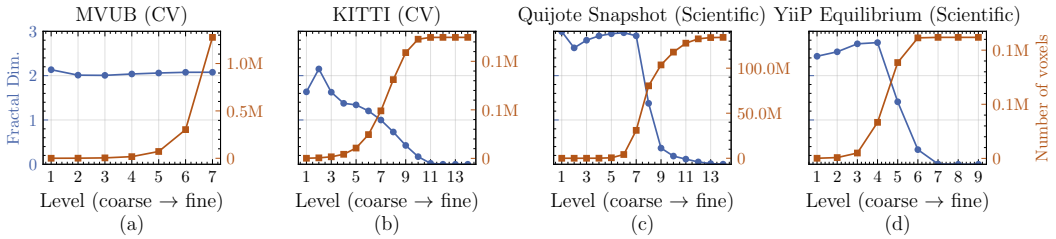


Figure 1: **Fractal dimension reveals a geometric difference between CV and scientific point clouds. (a) Human body scan.** The dimension remains near 2 at all levels, consistent with locally planar surface geometry. **(b) Vehicular LiDAR.** Fractal dimension increases from near 0 at fine levels to approximately 2 at coarse levels. This reflects the physical nature [18] of spinning LiDAR sensors: scan rings appear locally linear at finer levels and appear denser at coarser levels. **(c) Cosmology N-body simulations.** Fractal dimension rises quickly from 0 to 3, indicating a sharp transition from sparse to volume-filling geometry. **(d) Molecular dynamics simulations** share the sharp sparse-to-dense transition as in cosmology simulations. This contrast illustrates the fundamental geometric difference between CV point clouds and scientific particle data discussed in Sec. 2.1.

substantially higher compression ratios than G-PCC on standard computer vision benchmarks. These advances prompt a natural question: *can similar gains be realized for scientific particle datasets?* Comparable gains on scientific particle data would dramatically reduce storage and data movement costs, making large-scale particle simulations far more accessible for downstream analysis.

In this paper, we investigate the central question: ***Do neural compression methods achieve comparable effectiveness on scientific particle data as they do on computer vision point clouds?*** We further decompose it into three research questions: **RQ1:** *How does scientific particle data differ from computer vision point clouds?* **RQ2:** *Is the octree an efficient representation for scientific particle data?* **RQ3:** *Given the octree representation, do neural models improve over simple baselines?*

Guided by these questions, we provide a systematic analysis of where and why neural compression methods succeed or fail on particle data. We offer both benchmarking guidelines and practical perspectives for the community at the intersection of neural compression and scientific computing.

2 Analysis: Are existing neural compressors effective for particle data?

Our evaluation spans three scientific domains: cosmology, molecular dynamics, and plasma physics. For cosmology, we use the Quijote N-body simulations [39]. For molecular dynamics, we use the YiiP membrane-protein system [13] simulated with GROMACS. For plasma physics, we use the charged-particle beam dataset generated by WarpX [38] in Sandberg et al. [35]. Particle simulations store positions as floating-point numbers, whereas PCC codecs require integer-valued voxel coordinates. We apply uniform scalar quantization $x_i = \lfloor x/\Delta \rceil$, choosing Δ at or below the physical resolution limit of each domain, ensuring that quantization error is negligible for downstream analyses.

2.1 How is scientific particle data different from CV point clouds?

CV point clouds share a common geometric property: From CG models [7], LiDAR scans [18, 32], to RGB-D captures [11, 8, 30], all CV point clouds describe object surfaces, and are locally planar. Scientific particles do not share this property. Particles in scientific simulations represent different physical entities depending on the domain, and their spatial distribution vary greatly.

We demonstrate this distinction through the lens of *fractal dimension* [12], which characterizes how a point cloud fills space across scales. Consider partitioning the bounding volume into a grid of $2^l \times 2^l \times 2^l$ cells and marking each cell as occupied if it contains at least one point. Let N_l denote the number of occupied voxels. The fractal dimension, computed by $d_l = \log_2(N_l/N_{l-1})$, is a metric that captures how quickly the number of occupied voxels grows as grid resolution doubles. While this metric need not be an integer, it reduces to the more familiar Euclidean dimension in regular cases: it evaluates to 3 for solid volumes, 2 for surfaces, and 1 for lines. Due to its coincidence with Euclidean dimension and ease of computation, it has been used as an indicator of point sparsity in the PCC literature [33, 40].

Our results are shown in Fig. 1. CV point clouds representing surfaces have a consistent fractal dimension close to 2. Scientific particle data, by contrast, exhibits different patterns:

Table 1: **Neural methods achieve far smaller gains on scientific particle data than on CV data.** Bit-per-point (bpp) by method, with compression ratio over post-quantization bitrate. General-purpose scientific compressor SZ3 suffers on particle data, while particle compressor LCP matches the compression ratio of CV point cloud compressors. "–" marks entries where the method was not evaluated on the dataset originally.

	Dataset (# test samples)	Post-Quant.	Raw Octree	G-PCC	RENO	OctAttention	SZ3 [26]	LCP [42]
Scientific	Quijote, halos (200)	54	87.40 (0.6×)	35.92 (1.5×)	35.55 (1.5×)	35.78 (1.5×)	60.18 (0.9×)	36.22 (1.5×)
	Quijote, particles (200)	54	88.37 (0.6×)	36.03 (1.5×)	35.77 (1.5×)	35.92 (1.5×)	59.46 (0.9×)	36.25 (1.5×)
	Quijote, boxes (1600)	45	53.89 (0.8×)	23.52 (1.9×)	22.68 (2.0×)	22.82 (2.0×)	40.36 (1.1×)	24.52 (1.8×)
	YiiP (900)	45	59.70 (0.8×)	25.35 (1.8×)	22.89 (2.0×)	23.91 (1.9×)	44.70 (1.0×)	27.08 (1.7×)
	WarpX (30)	54	55.84 (1.0×)	23.74 (2.3×)	23.35 (2.3×)	23.33 (2.3×)	41.25 (1.3×)	24.70 (2.2×)
CV	MVUB, Phil (245)	30	2.55 (11.8×)	1.07 (28.0×)	–	0.79 (38.0×)	–	–
	8iVFB, Loot (300)	30	2.98 (10.1×)	0.95 (31.6×)	–	0.62 (48.4×)	–	–
	KITTI (7518)	52	17.85 (2.9×)	8.95 (5.8×)	7.32 (7.1×)	–	–	–
	Ford (60)	52	24.56 (2.1×)	11.72 (4.4×)	9.75 (5.3×)	–	–	–

1. **Cosmology N-body** simulations [37] track how dark matter particle interact under gravity. Over time, particles collapse into a complex web of voids, filaments, sheets, and dense halos [4]. These structures span a wide range of intrinsic dimensionalities (See Fig. 1 (c)).
2. **Molecular dynamics** simulations [1, 21] track atom movements in biomolecular systems such membranes. Atomic positions are governed by interactions at distinct scales: intramolecular forces such as covalent bonds constrain atoms into rigid geometries, while intermolecular forces drive the arrangement of molecules relative to each other (See Fig. 1 (d)).
3. **Plasma physics** [14] model charged particles in electromagnetic fields, producing distributions shaped by beam dynamics and instabilities that, again, do not have surface-like structure.

2.2 Is octree an efficient representation for scientific particle data?

Octree is the most common way to represent CV point cloud data for compression [36, 20, 16]. It represents a point cloud by recursively splitting the bounding volume into eight sub-volumes, encoding at each node an 8-bit occupancy code that signals which children are occupied. The problem of compressing plain coordinates $\{x_i, y_i, z_i\}$ is converted to compressing the breadth-first traversal of occupancy code sequence $\{O_k\}$.

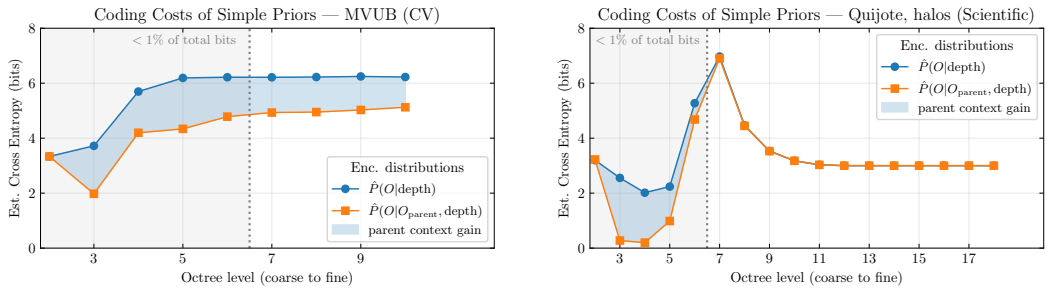
The octree representation offers three advantages: multi-resolution decoding [24], representation-level compression, and support for context-adaptive entropy coding on occupancy code sequence. For a point cloud with consistent voxel merging from a fine level to a coarse level, the cost of occupancy codes is easily offset by the rapidly reducing number of octree nodes [5, 25]. Botsch et al. [5] showed that for a consistently planar point cloud, the bit-per-point cost of octree is upper-bounded by 2.67.

This advantage diminishes for scientific particle data: d exceeds 2 at coarse levels but rapidly approaches 0. At d close to 0, octree is no longer efficient, as it achieves data reduction by merging voxels. When no voxels merge, an 8-bit occupancy signals at most $\log_2 8 = 3$ bits of information. 5 bits are wasted on each point, and this waste accumulates over multiple levels. This waste can be partly recovered by hand-designed flags [34] or learned entropy coding. Still, the inefficiency suggests that octree, despite being predominant for CV, may not suit scientific particle data.

2.3 Do neural codecs improve particle data compression?

To isolate the benefit of learned context models for octree coding, we train two neural compression models on curated multi-frame training datasets from three domains. For Quijote, we derive three variants, halos, particles, and boxes. They are all drawn from 2000 simulations at redshift zero with varying cosmological parameters through Latin hypercube. The halo and particle frames are subsampled to 100,000 points to fit GPU memory. For YiiP, we use a 9001-frame trajectory with 111,815 atoms per frame. For the WarpX dataset, we partition 1 million particle positions at each of 2 iterations across 15 accelerator stages into 300 frames of 100,000 points each. After quantization and duplicate removal, each dataset provides a collection of integer-coordinate point clouds.

The results reported in Table 1 show that neural compression models OctAttention [15] and RENO [41] achieve surprisingly small additional gains over G-PCC. Across scientific datasets, neural compression models achieve at best 10% bitrate reduction over G-PCC (RENO on YiiP), as opposed to around 16% on LiDAR and 30% on human scans.



(a) (b)

Figure 2: **CV point cloud compression benefits across levels from conditioning on a single parent node, while particle datasets only benefits from it at coarser levels. (a)** On MVUB (human body scans), points concentrate on surfaces, so a child's occupancy can be strongly predicted by its parent's: conditional entropy drops significantly across levels. **(b)** Quijote particle data benefits from the same model only on coarser levels. Gains in these levels are less significant, because the number of voxels to code here is negligible compared to the entire octree. Gray area indicates levels that contribute less than 1% of the total bitstream – gains in these levels have little effect on overall bpp.

To build an intuition for why neural context models yield only marginal gains, we examine the core problem of compressing occupancy code sequence $\{O_k\}$ through a simple but informative test: conditioning on the parent node. In Fig. 2, we illustrate how a simple context-based model is effective on CV point clouds but offers less gain on scientific particles. We start with a simple context model based on level marginals: $\hat{P}(O|\text{depth})$, and examine the coding gains achieved using an encoding distribution $\hat{P}(O|O_{\text{parent}}, \text{depth})$, which has a parent occupancy code as additional context. In Fig. 2, we observe that conditioning on parent occupancy code can significantly reduce coding costs for CV dataset MVUB at almost all levels, but it only mildly helps for scientific particle data Quijote.

3 Perspectives: How can learning help scientific particle data compression in the future?

Neural position quantization This work follows the common practice of uniformly quantizing position data to the point that the loss is negligible for certain downstream analyses. This is straightforward, but may not be optimal. Uniform quantization ignores that the importance of spatial regions can be non-uniform. A promising direction of machine learning is to replace uniform quantization with adaptive quantization that still preserves physically meaningful information. For example, topology-preserving compressors such as TopoSZp [2] shows that spatially adaptive quantization can preserve critical points in data while allowing aggressive bitrate savings.

Neural spatial partition and traversal The octree imposes a uniform 8-way split at every node and a hard-coded traversal order. Predictive codecs in neural PCC consume occupancy codes from the most significant bits of the coordinates to the least, rather than following the native storage order. Both choices for partitioning and traversal order are practical for CV point clouds but mismatched to the geometries of scientific data; a promising direction is to learn both in a data-driven manner.

Leveraging neural emulators as learned priors Neural emulators for physical simulations can be repurposed as a learned prior for lossy compression. Emulators such as Jamieson et al. [22] learn to map low-dimensional inputs (initial conditions and physical parameters) to full particle distributions. It can be used as a physics-aware prior during predictive coding. This framework generalizes to more domains such as MD force field [3] and plasma [19].

Bridging the CV and HPC communities Progress on scientific particle compression will benefit from collaborations between two communities that work in parallel. CV community brings sophisticated methods for 3D scene understanding and compression. HPC community has deep expertise in the domain knowledge: what are the physical properties that must be preserved, what is a tolerable distortion, how are data generated in supercomputers, and how compression should fit in the data management pipeline. Concrete directions for collaboration include joint benchmarks that span both domains, co-designed codecs that are informed by physics of the domain.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science Foundation (NSF) under grant number 2341055. This work was also supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR), under contract DE-AC02-06CH11357. This work was also supported by the U.S. National Science Foundation (OAC 2311875, OAC 2514036).

References

[1] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C Smith, Berk Hess, and Erik Lindahl. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25, 2015.

[2] Tripti Agarwal, Sheng Di, Xin Liang, Zhaoyuan Su, Yuxiao Li, Ganesh Gopalakrishnan, Hanqi Guo, and Franck Cappello. Toposzp: Lightweight topology-aware error-controlled compression for scientific data. *arXiv preprint arXiv:2602.17552*, 2026.

[3] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1): 2453, 2022.

[4] J Richard Bond, Lev Kofman, and Dmitry Pogosyan. How filaments of galaxies are woven into the cosmic web. *Nature*, 380(6575):603–606, 1996.

[5] Mario Botsch, Andreas Wiratanaya, and Leif Kobbelt. Efficient high quality rendering of point sampled geometry. In *13th Eurographics Workshop on Rendering: Pisa, Italy, June 26-28, 2002*, page 53. Association for Computing Machinery, 2002.

[6] Franck Cappello, Allison Baker, Ebru Bozda, Martin Burtscher, Kyle Chard, Sheng Di, Paul Christopher O Grady, Peng Jiang, Shaomeng Li, Erik Lindahl, et al. Lossy compression of scientific data: Applications constrains and requirements. *arXiv preprint arXiv:2503.20031*, 2025.

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[9] Xinyue Dai, Hongxia Zhou, and Xinying Wei. Molecular dynamics simulation for deposition and bonding mechanisms of ti particles upon impact in cold spray. *Langmuir*, 41(18):11428–11442, 2025.

[10] Marco De Vivo, Matteo Masetti, Giovanni Bottegoni, and Andrea Cavalli. Role of molecular dynamics and related methods in drug discovery. *Journal of medicinal chemistry*, 59(9): 4035–4061, 2016.

[11] Eugene d'Eon, Bob Harrison, Taos Myers, and Philip A. Chou. 8i voxelized full bodies – a voxelized point cloud dataset. In *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006*, Geneva, 2017.

[12] Kenneth Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. John Wiley & Sons, 2nd edition, 2003. ISBN 978-0470848616.

[13] Shujie Fan and Oliver Beckstein. Molecular dynamics trajectory of membrane protein YiiP, 2019.

[14] Luca Fedeli, Axel Huebl, France Boillod-Cerneux, Thomas Clark, Kevin Gott, Conrad Hillairet, Stephan Jaure, Adrien Leblanc, Rémi Lehe, Andrew Myers, et al. 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*, pages 1–12. IEEE, 2022.

[15] Chunyang Fu, Ge Li, Rui Song, Wei Gao, and Shan Liu. Octattention: Octree-based large-scale contexts model for point cloud compression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 625–633, 2022.

[16] Wei Gao, Liang Xie, Songlin Fan, Ge Li, Shan Liu, and Wen Gao. Deep learning-based point cloud compression: An in-depth survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[17] Lehman H Garrison, Daniel J Eisenstein, Douglas Ferrer, Nina A Maksimova, and Philip A Pinto. The abacus cosmological N-body code. *Monthly Notices of the Royal Astronomical Society*, 508(1):575–596, 09 2021. ISSN 0035-8711. doi: 10.1093/mnras/stab2482. URL `https://doi.org/10.1093/mnras/stab2482`.

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

[19] Vignesh Gopakumar, Stanislas Pamela, Lorenzo Zanisi, Zongyi Li, Ander Gray, Daniel Brennand, Nitesh Bhatia, Gregory Stathopoulos, Matt Kusner, Marc Peter Deisenroth, et al. Plasma surrogate modelling using fourier neural operators. *Nuclear Fusion*, 64(5):056025, 2024.

[20] Danillo Graziosi, Ohji Nakagami, Satoru Kuma, Alexandre Zaghetto, Teruhiko Suzuki, and Ali Tabatabai. An overview of ongoing point cloud compression standardization activities: Video-based (v-pcc) and geometry-based (g-pcc). *APSIPA Transactions on Signal and Information Processing*, 9:e13, 2020.

[21] Scott A Hollingsworth and Ron O Dror. Molecular dynamics simulation for all. *Neuron*, 99(6): 1129–1143, 2018.

[22] Drew Jamieson, Yin Li, Renan Alves de Oliveira, Francisco Villaescusa-Navarro, Shirley Ho, and David N Spergel. Field-level neural network emulator for cosmological n-body simulations. *The Astrophysical Journal*, 952(2):145, 2023.

[23] Martin Karplus and J Andrew McCammon. Molecular dynamics simulations of biomolecules. *Nature structural biology*, 9(9):646–652, 2002.

[24] Aaron Knoll. A survey of octree volume rendering methods. *VLUDS*, pages 87–96, 2006.

[25] Sébastien Lasserre, David Flynn, and Shouxing Qu. Using neighbouring nodes for the compression of octrees representing the geometry of point clouds. In *Proceedings of the 10th ACM Multimedia Systems Conference*, pages 145–153, 2019.

[26] Xin Liang, Kai Zhao, Sheng Di, Sihuan Li, Robert Underwood, Ali M Gok, Jiannan Tian, Junjing Deng, Jon C Calhoun, Dingwen Tao, et al. SZ3: A modular framework for composing prediction-based error-bounded lossy compressors. *IEEE Transactions on Big Data*, 9(2): 485–498, 2022. URL `https://doi.org/10.1109/TBDATA.2022.3201176`.

[27] Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

[28] Peter Lindstrom. Fixed-rate compressed floating-point arrays. *IEEE transactions on visualization and computer graphics*, 20(12):2674–2683, 2014.

[29] Nina A Maksimova, Lehman H Garrison, Daniel J Eisenstein, Boryana Hadzhiyska, Sownak Bose, and Thomas P Satterthwaite. AbacusSummit: a massive set of high-accuracy, high-resolution N-body simulations. *Monthly Notices of the Royal Astronomical Society*, 508 (3):4017–4037, 09 2021. ISSN 0035-8711. doi: 10.1093/mnras/stab2484. URL `https://doi.org/10.1093/mnras/stab2484`.

[30] Khaled Mammou, Philip A. Chou, David Flynn, Maja Krivokuća, Ohji Nakagami, and Toshiyasu Sugio. G-PCC codec description v2. Technical Report N18189, ISO/IEC JTC1/SC29/WG11 MPEG, Marrakech, 2019.

[31] MPEG 3D Graphics WG 7 and Haptics Coding. G-pcc 2nd edition codec description. Technical report, ISO/IEC JTC 1/SC 29/WG 7, 2023.

[32] Gaurav Pandey, James R McBride, and Ryan M Eustice. Ford campus vision and lidar data set. *The International Journal of Robotics Research*, 30(13):1543–1552, 2011.

[33] Jiahao Pang, Muhammad Asad Lodhi, and Dong Tian. Grasp-net: Geometric residual analysis and synthesis for point cloud compression. In *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis*, pages 11–19, 2022.

[34] Jingliang Peng and CC Jay Kuo. Octree-based progressive geometry encoder. In *Internet Multimedia Management Systems IV*, volume 5242, pages 301–311. SPIE, 2003.

[35] Ryan Sandberg, Remi Lehe, Chad Mitchell, Marco Garten, Andrew Myers, Ji Qiang, Jean-Luc Vay, and Axel Huebl. Synthesizing particle-in-cell simulations through learning and gpu computing for hybrid particle accelerator beamlines. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pages 1–11, 2024.

[36] Ruwen Schnabel and Reinhard Klein. Octree-based point-cloud compression. *PBG@ SIGGRAPH*, 3(3), 2006.

[37] Volker Springel. The cosmological simulation code gadget-2. *Monthly notices of the royal astronomical society*, 364(4):1105–1134, 2005.

[38] Jean-Luc Vay, Marco Acciarri, Ann Almgren, Lígia Diana Amorim, Igor Andriyash, Justin Ray Angus, Thierry Antoun, Daniel Belkin, David Bizzozero, Aurore Blelly, Stephen Eric Clark, Eya Dammak, Luca Fedeli, Arianna Formenti, Marco Garten, Lixin Ge, Lorenzo Giacomel, Kevin Gott, Lorenzo Giacomel, Roelof E. Groenewald, David Grote, Junmin Gu, Cyrus Harrison, Muhammad Haseeb, Axel Huebl, Revathi Jambunathan, Ilian Kara-Mostefa, Hannah Klion, Prabhat Kumar, Remi Lehe, Burlen Loring, Thomas Marks, Phil Miller, Andrew Myers, Juliette Pech, Elisa Rheaume, Glenn Richardson, Elisa Rheaume, Michael E. Rowan, Ryan Thor Sandberg, Peter Scherpelz, Alexander Sinn, Olga Shapoval, Davide Terzani, Maxence Thévenet, Kale Weichman, Eloise Yang, Neïl Zaim, Weiqun Zhang, Yinjian Zhao, Kevin Z. Zhu, and Edoardo Zoni. WarpX: An advanced Particle-In-Cell code. https://blast-warpx.github.io, 2018. URL `https://doi.org/10.5281/zenodo.4571577`. `https://doi.org/10.5281/zenodo.4571577`.

[39] Francisco Villaescusa-Navarro, ChangHoon Hahn, Elena Massara, Arka Banerjee, Ana Maria Delgado, Doogesh Kodi Ramanah, Tom Charnock, Elena Giusarma, Yin Li, Erwan Allys, Antoine Brochard, Cora Uhlemann, Chi-Ting Chiang, Siyu He, Alice Pisani, Andrej Obuljen, Yu Feng, Emanuele Castorina, Gabriella Contardo, Christina D. Kreisch, Andrina Nicola, Justin Alsing, Roman Scoccimarro, Licia Verde, Matteo Viel, Shirley Ho, Stephane Mallat, Benjamin Wandelt, and David N. Spergel. The Quijote Simulations. *Astrophysical Journal, Supplement*, 250(1):2, September 2020. doi: 10.3847/1538-4365/ab9d82.

[40] Jianqiang Wang, Ruixiang Xue, Jiaxin Li, Dandan Ding, Yi Lin, and Zhan Ma. A versatile point cloud compressor using universal multiscale conditional coding–part i: Geometry. *IEEE transactions on pattern analysis and machine intelligence*, 47(1):269–287, 2024.

[41] Kang You, Tong Chen, Dandan Ding, M Salman Asif, and Zhan Ma. Reno: Real-time neural compression for 3d lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22172–22181, 2025.

[42] Longtao Zhang, Ruoyu Li, Congrong Ren, Sheng Di, Jinyang Liu, Jiajun Huang, Robert Underwood, Pascal Grosset, Dingwen Tao, Xin Liang, et al. LCP: Enhancing scientific data management with l ossy c ompression for p articles. *Proceedings of the ACM on Management of Data*, 3(1):1–27, 2025.

[43] Kai Zhao, Sheng Di, Danny Perez, Xin Liang, Zizhong Chen, and Franck Cappello. Mdz: An efficient error-bounded lossy compressor for molecular dynamics. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 27–40, 2022. doi: 10.1109/ICDE53745.2022.00007.

A Comparison with Scientific Compressors

In this section, we report compression results from two representative scientific compressors. SZ3 [26] is a general-purpose error-bounded lossy compressor that combines interpolation- and prediction-based coding. It is best suited for multi-dimensional field data, where values vary smoothly in a neighborhood of grid points. LCP [43], in contrast, is an error-bounded compressor designed specifically for scientific particles. It re-orders and groups points by their spatial location to form small local blocks. This induces local correlation within each block and yields superior compression ratio [43] than general-purpose scientific compressors.

The results reported in Table 1 show that general-purpose scientific compressor SZ3 performs poorly. In some cases (Quijote halos and particles), SZ3 suffers from extremely large prediction errors on some points, resulting in the compressor applying lossless compression directly on the floating-point number. This results in an overall larger compressed size (compression ratio < 1) than the post-quantization bpp. In contrast, LCP, a dedicated scientific particle compressor, achieves compression ratio comparable to CV codecs. It benefits from the freedom of re-ordering and grouping points based on their spatial positions.

B Scientific Dataset Curation

Both octree-based codecs such as G-PCC and neural point cloud codecs operate on integer-valued 3d voxels, while scientific particle simulations store position data as floats. Applying existing point cloud codecs to scientific data requires us to bridge the gap from continuous coordinates to discrete grid positions. Across all the dataset, we adopt a near-lossless preconditioning through uniform scalar quantization. A continuous coordinate x is mapped to quantization index $x_i = \lfloor x/\Delta \rceil$. At de-quantization, quantization index is mapped back to continuous coordinate by $\hat{x} = \Delta \cdot x_i$. The quantization step size Δ is tailored to the requirement of the scientific domain, ensuring that the quantization error remains negligible for downstream analyses.

Cosmology N-body Simulations The Quijote simulations [39] is a large scale cosmology N-body simulation dataset designed to enable training of machine learning models for cosmology. It contains more than 40 thousands simulations under various physical constraints and spatial resolutions. In order to cover different kinds of geometries, we derive 3 different particle datasets from the Quijote simulations: **halos**, **particles**, and **boxes**. All 3 variants are derived from 2000 simulations at redshift zero. Each simulation tracks the evolution of 512^3 particles in a volume of $(1\,h^{-1}\,\mathrm{Gpc})^3$ with periodic boundary condition. These 2000 simulations differ in 5 cosmological parameters, which were generated through Latin hypercube sampling.

The **halos** dataset is a collection of subsampled dark matter halo position data. We read halo positions from the Gadget-format HDF5 files and convert to coordinates in $h^{-1}\,\mathrm{Mpc}$. From each snapshot, we uniformly subsample 100,000 halos at random using a per-realization random seed.

The **particles** dataset is a collection of subsampled dark matter position data. Similar to the **halos** dataset, we read particle positions from the Gadget-format HDF5 files and converting to coordinates in $h^{-1}\,\mathrm{Mpc}$. From each snapshot, we uniformly subsample 100,000 particles at random using a per-realization random seed.

The **boxes** dataset is a collection of particles inside a small cubic volume. From the same 2000 snapshots used to generate **halos** and **particles**, we extract particles within a small cubic block of volume $(125\,h^{-1}\,\mathrm{Mpc})^3$. We extract 8 randomly sampled blocks from each snapshot. This dataset is introduced to observe the the compression performance of finer details of cosmology simulation, as opposed to **halos** and **particles**, which retain only larger scale structures.

For all three variants of Quijote data, we apply uniform scalar quantization to the positions as a near-lossless preconditioning prior to compression. The step size of $\Delta = 7.6\,h^{-1}\,\mathrm{Kpc}$ is chosen to be well below the gravitational softening length of the simulation. The softening length [37] is the spatial scale below which gravitational forces are intentionally smoothed to prevent numeric explosion in close particle encounters. Therefore, no meaningful physical structure is resolved beneath this scale during simulation. Consequently, the information discarded by a quantization step even smaller than the gravitational softening length is effectively lossless for downstream analyses.

Molecular Dynamics The YiiP equilibrium dataset [13] contains molecular dynamic (MD) trajectories of a membrane-protein system. It tracks the coordinate data of all the entire system, including protein, membrane, water, and ions. Coordinate data are stored per atom for 111815 atoms in 2 trajectories with 901 and 9001 frames, respectively. In order to acquire enough data to train neural compression models, we use the longer trajectory with 9001 frames.

The membrane-protein system was simulated with high performance MD simulation code GROMACS [1], and coordinate data are stored with 0.001 nm resolution. For this data, we apply quantization step of $\Delta = 0.001$ nm, which is the same as the storage resolution.

Plasma Physics Sandberg et al. [35] proposed a dataset of charged particle beams in plasma accelerator. The dataset tracks 1 million particles at 2 iterations, representing the initial and final snapshots. For each iteration, the particle position is recorded separately for 15 accelerator stages. We sequentially partition that 1 million positions at each stage into 100,000 point chunks, and create a dataset with 300 frames with 100, 000 points each. We quantize the positions with $\Delta = 1$ nm, which is small enough to results in less than 0.01% point merging.