UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Learning-based Point Cloud Geometry Compression with Transformers and Scalable Coding

by

Zhe Luo

A Thesis Submitted
in Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2026

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Zhe Luo, declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research was supported by an Australian Government Research Training Program (RTP) Scholarship doi.org/10.82133/C42F-K220.

Signature: Production Note:
Signature removed prior to publication.

Date: March-01-2026

ABSTRACT

Point clouds, as unstructured collections of 3D points, have become essential for representing complex geometries in applications ranging from autonomous driving to immersive media and cultural heritage preservation. However, their large data and irregular structures pose significant challenges in storage and transmission, motivating research into advanced compression techniques that balance rate-distortion (RD) performance while supporting scalable decoding and compressed-domain analysis. This thesis addresses these challenges through proposed learning-based point cloud compression (PCC) frameworks. Firstly, this thesis proposes a transformer-based autoencoder with local neighbor aggregation and global attention mechanisms to preserve intricate local and global features, achieving superior geometry coding and BD-Rate performance over recent baseline methods on ShapeNetCorev2. Secondly, this thesis introduces an edge-preserving compressed-domain classification approach using local attention to extract features directly from bitstreams, enhancing classification accuracy on ModelNet40 at reduced bitrates. Third, this thesis develops ScalaDAT, a density-aware scalable PCC method employing a tail-drop strategy for incremental decoding in a single training iteration, enhancing RD efficiency on widely-used datasets like SemanticKITTI and ShapeNet, outperforming benchmarks. This thesis's contributions advance PCC by integrating and enhancing transformer architectures, attention-driven feature preservation, and scalable mechanisms, enabling scalable, efficient 3D data handling for real-life applications. These advancements not only mitigate computational and bandwidth constraints but also pave the way for seamless integration with downstream tasks, fostering innovations in 3D spatial data processing.

Dissertation directed by Prof. Stuart Perry and A/Prof. Wenjing Jia
School of Electrical and Data Engineering

Dedication

This thesis is dedicated to my resilient self, who persevered tirelessly through immense pressure without yielding and improving myself; to my parents, whose steadfast support has been my beacon; and to my lifelong mentor, Mr. Yang Shen, whose wisdom and generosity have profoundly shaped my path.

Acknowledgements

First and foremost, I would like to acknowledge my supervisors, Prof. Stuart Perry and A/Prof Wenjing Jia, for their continuous guidance, insightful feedback, and unwavering encouragement throughout my PhD journey.

I am deeply grateful to my parents and uncle for their endless support, assistance, and motivation that sustained me through challenging times.

A special thanks to my cats, DotDot and ThoughtThought, for their companionship, daily comfort, and unconditional love.

Finally, I extend my appreciation to all those who contributed directly or indirectly to my research, including JPEG researchers, mentors, SEDE and FEIT members, and the authors of the cited papers.

Zhe Luo
Sydney, Australia, 2026.

List of Publications

Journal Papers

1. Z. Luo, W. Jia, and S. Perry, "Compressed point cloud classification with point-based edge sampling," *EURASIP Journal on Image and Video Processing*, vol. 2024, no. 1, p. 18, 2024.

Conference Papers

1. J. Prazeres, Z. Luo, A. M. G. Painheiro, L. A. da Silva Cruz, and S. Perry, "JPEG pleno call for proposals responses quality assessment," in *ICASSP 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

2. Z. Luo, W. Jia, and S. Perry, "Transformer-based geometric point cloud compression with local neighbor aggregation," in *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2023, pp. 223–228.

Submitted Manuscripts

1. Z. Luo, W. Jia, and S. Perry, "ScalaDAT: Scalable density-aware tail-drop for point cloud compression," *submitted for publication.*

Contents

List of Figures

List of Tables

Abbreviation

- AI - Artificial Intelligence
- APES - Attention-based Point cloud Edge Sampling
- AR - Augmented Reality
- BD-Rate - Bjøntegaard Delta Rate
- BD-Top-1 - Bjøntegaard Delta Top-1 Accuracy
- BPP - Bits per Point
- CAD - Computer-Aided Design
- CD - Chamfer Distance
- CNN - Convolutional Neural Network
- CPCC - Compressed Point Cloud Classification
- CPCA - Compressed Point Cloud Analysis
- DPCA - Decompressed Point Cloud Analysis
- DPCC - Decompressed Point Cloud Classification
- EMA - Exponential Moving Average
- FPS - Farthest Point Sampling
- GIS - Geographic Information Systems
- G-PCC - Geometry-based Point Cloud Compression
- IB - Information Bottleneck

- IRN - Inception Residual Network
- JPEG - Joint Photographic Experts Group
- KNN - K-Nearest Neighbors
- LAM3D - Large Image-Point-Cloud Alignment Model
- LiDAR - Light Detection and Ranging
- LNA - Local Neighbor Aggregation
- LoD - Level-of-Detail
- LPCCC - Learned Point Cloud Compressed Classification
- MAE - Masked Autoencoder
- MLP - Multilayer Perceptron
- MPEG - Moving Picture Experts Group
- MR - Mixed Reality
- MSE - Mean Squared Error
- MVUB - Microsoft Voxelized Upper Bodies
- NN - Nearest Neighbor
- PCA - Point Cloud Analysis
- PCC - Point Cloud Compression
- PCT - Point Cloud Transformer
- PES - Point-based Edge Sampling
- ProgDTD - Progressive Double-Tail-Drop
- PSNR - Peak Signal-to-Noise Ratio
- PSNR-D1 - Peak Signal-to-Noise Ratio Distance1

- PSNR-D2 - Peak Signal-to-Noise Ratio Distance2
- PST-NET - Point Sampling Transformer Network
- PT - Point Transformer
- RAHT - Region Adaptive Hierarchical Transform
- RD - Rate-Distortion
- REPS - Region-Adaptive Edge-Preserving Sampling
- RGB - Red Green Blue
- RNN - Recurrent Neural Network
- ROI - Region of Interest
- ScalaDAT - Scalable Density-Aware Tail-drop
- SPCC - Scalable Point Cloud Compression
- SR - Scalable Ratio
- SVC - Scalable Video Coding
- trisoup - Triangular Soup
- V-PCC - Video-based Point Cloud Compression
- VM - Verification Model
- VR - Virtual Reality
- XR - Extended Reality
- YOGA - Yet anOther Geometry-based point cloud codec with Attributes

Chapter 1

Introduction

Point clouds represent 3D spatial data as collections of discrete points defined by their coordinates in a 3D space, often augmented with attributes such as colour, intensity, or density. This unstructured yet versatile format enables high-fidelity capture of the geometry and appearance of physical objects and environments, making point clouds indispensable in diverse fields, including autonomous systems, robotics, urban planning, cultural heritage digitisation, and immersive media [3, 4]. The proliferation of advanced acquisition technologies—such as Light Detection and Ranging (LiDAR), photogrammetry, and depth-sensing—has facilitated the creation of large-scale, high-resolution datasets [5, 6], while concurrent computational advancements have improved processing and analysis capabilities [7]. However, the inherent challenges of point clouds, including their irregularity, sparsity, and substantial data volume, impose significant burdens on storage and transmission [8]. These limitations underscore the need for effective representation (*e.g.*, voxel-based), analysis, and compression strategies to enhance efficiency without compromising fidelity.

This research is driven by the increasing demand for robust point cloud compression (PCC) in bandwidth-constrained applications like extended reality (XR), autonomous driving, and scene reconstruction, where data volume requirements exacerbate inefficiencies [9]. Existing methods fall short in preserving fine details, maintaining accuracy at low bitrates, and supporting scalable decoding, often resulting in artefacts and degraded performance [10, 11, 12].

This thesis focuses on developing learning-based solutions that leverage Trans-

formers and attention mechanisms to achieve superior rate-distortion (RD) performance, enhanced compressed-domain classification, and efficient Scalable geometry coding. Key achievements include notable improvements in RD performance (measured via BD-Rate), enhanced low-bitrate accuracy for downstream tasks (measured via BD-Top-1 Accuracy), and a scalable PCC framework that demonstrates competitive coding efficiency against recent baseline methods on benchmark datasets.

To provide a structured overview, this chapter is organised as follows:

- **Section 1.1** explores the background and motivations for point cloud compression.
- **Section 1.2** defines the core problem and associated challenges.
- **Section 1.3** outlines the research objectives.
- **Section 1.4** highlights the thesis contributions.
- **Section 1.5** details the overall thesis organisation.

1.1 Background and Motivation

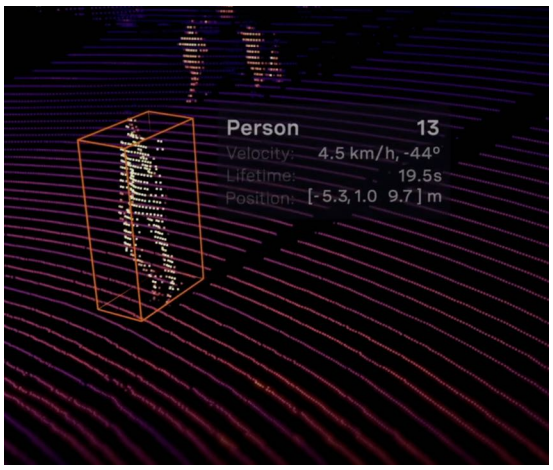
1.1.1 Background: Importance of Point Cloud

Point cloud data has emerged as a pivotal representation in 3D modelling, capturing spatial information through discrete points in a coordinate system, often enriched with attributes, such as colour, intensity, or normals. This format's versatility stems from its ability to depict complex geometries without the constraints of mesh-based connectivity, enabling seamless integration across diverse applications [13]. The importance of point cloud data is particularly pronounced in immersive technologies such as augmented reality (AR), virtual reality (VR), and mixed reality (MR), where it facilitates the creation of realistic 3D environments for enhanced

user interaction and immersion [14, 15]. For instance, in VR/AR applications, point clouds support telepresence systems and volumetric video, enabling virtual replicas and gesture-based interactions, with datasets like MVUB [16] and 8iVFB.v2 [17] used for rendering and compression in immersive avatar experiences.

To explicitly illustrate the primary application domains targeted in this research, Figure 1.1 visualises two typical scenarios where effective point cloud compression is essential.

(a) High-fidelity 3D reconstruction



(b) Sparse spatial data streaming

Figure 1.1 : Typical application scenarios motivating research into advanced point cloud compression: (a) High-fidelity 3D asset and scene archival (*e.g.*, digital twins and dense building scans) requiring rigorous structural preservation; (b) Bandwidth-constrained streaming of sparse spatial data (*e.g.*, smart city monitoring and remote surveillance) requiring scalable data transmission. Images adapted from Alterpex [18] and Ouster [19].

Furthermore, point clouds enable high-fidelity 3D reconstructions for remote collaboration, simulating physical presence in virtual spaces [20]. Beyond immersive media, applications extend to medical imaging for detailed anatomical modelling, cultural heritage preservation through digital archiving [21], and geographic infor-

mation systems (GIS) for large-scale environmental monitoring [22]. In remote sensing, point clouds are vital for urban land cover classification, shoreline mapping, and landslide assessment, with datasets like Semantic3D [23] supporting tasks in precision mapping.

Additionally, point clouds are instrumental in indoor scene understanding and spatial computing, utilising datasets such as ScanNet [24] to map complex environments. Rather than targeting latency-critical vehicular navigation or autonomous driving, this thesis firmly positions its focus on scenarios motivating research into genuine transmission and storage: the high-fidelity archival of dense 3D objects and the bandwidth-constrained streaming of spatial environments.

Overall, point cloud data stands as a cornerstone in modern 3D technologies, offering unparalleled flexibility for capturing and processing complex spatial information across both academic and industrial domains [25].

1.1.2 Motivation: Why Point Cloud Compression?

Building on the above discussion, point clouds are pivotal for advancing 3D technologies, yet their massive data volumes pose substantial challenges in storage, transmission, and processing. The proliferation of high-resolution 3D sensors, including LiDAR and depth cameras, has enabled detailed point cloud models across domains like autonomous driving, immersive AR/VR, and cultural heritage preservation, but this has led to an exponential surge in data size [26, 3]. For example, high-quality point clouds often contain millions to billions of points, each with geometric coordinates and attributes like colour or intensity, resulting in gigabyte-scale files. For example, SemanticKITTI [5] encompasses approximately 4.5 billion points across sequences, while urban remote sensing scans (*e.g.*, SensatUrban [27]) can span hundreds of millions of points over vast areas, intensifying storage and bandwidth demands [28].

Compounding these issues is the inherent irregularity and sparsity of point clouds, where points are unevenly distributed and confined to object surfaces, leaving most 3D space unoccupied [29]. In the ShapeNet [6] dataset, for instance, one model can be sampled into thousands of points per shape, exhibiting occupancy rates below 1% when voxelised (*e.g.*, 0.8% for 16,384 points in a 2 million-voxel grid for one point cloud model), highlighting the inefficiency of naive representations. This sparsity, combined with variable density, complicates efficient handling and exacerbates overheads in bandwidth-constrained applications.

Unlike 2D images and videos, which benefit from structured grids and mature codecs like H.264 SVC [30], point clouds lack such frameworks, making compression essential to reduce data volume while preserving fidelity. Traditional PCC benchmarks from the Moving Picture Experts Group (MPEG), including Geometry-based Point Cloud Compression (G-PCC) [31] and Video-based Point Cloud Compression (V-PCC) [32], provide foundational solutions but face persistent limitations that severely undermine their practicality [3]:

- **Storage and Transmission Bottlenecks**: Point cloud data, particularly from LiDAR sensors, can accumulate up to gigabytes per scan or sequence, overwhelming storage infrastructures and causing significant delays in data transmission, especially for applications requiring remote data access or long-term archival [33].The high-dimensional nature and intrinsic complexity of point clouds exacerbate these issues, making efficient PCC methods essential for reducing memory usage and enabling faster transmission in applications like telepresence or remote sensing [10].

- **Degradation of Reconstruction Quality and Downstream Task Performance**: Inadequate handling of point cloud data can erode fine details through artefacts, compromising visual realism and leading to drawbacks in

the accuracy in downstream tasks such as 3D object detection [34]. In detail, these fidelity losses, including point displacement or occupancy inaccuracies, can degrade the accuracy of downstream tasks such as 3D object detection [34], underscoring the need for effective PCC techniques that preserve geometric integrity and support reliable application performance [3].

- **High Resource Demands Limiting Real-World Deployment**: Processing and compressing large-scale point cloud data demand substantial computational power and memory, restricting deployment on edge devices like mobile AR/VR headsets or embedded automotive systems [11]. This resource intensity not only increases energy costs but also impedes low-latency responsiveness, creating barriers to widespread adoption in constrained environments where low-latency, efficient compression is crucial for operational viability [35].

These profound challenges vividly illustrate the urgent demand for effective PCC methods to facilitate broader adoption in resource-limited settings while upholding data integrity and performance. Emerging learning-based PCC methods represent a transformative solution, harnessing neural networks—such as Autoencoders [36, 11, 37] and Transformers [38, 39]—to craft compact representations and superior RD optimisations that surpass traditional limitations [40]. These techniques excel at capturing intricate local and global geometric features, outperforming traditional codecs in reconstruction quality with a smaller bitstream. However, they still confront several challenges, including structural details preservation, high accuracy at low bitrates, and scalable coding. More details will be discussed in the next few sections.

1.2 Problem Definition

1.2.1 Definition: What is Point Cloud Compression?

Prior to examining the challenges in PCC, it is essential to define the concept precisely. PCC is an essential technique for minimising the storage and bandwidth demands of 3D point cloud data, which comprises unordered collections of points in a 3D space that model the geometry of objects or scenes, typically acquired via LiDAR sensors, depth cameras, or photogrammetry. This process mainly utilises three key point cloud representations—point-based, octree-based, and voxel-based—to structure the data effectively for handling sparsity, irregularity, and scale. Point-based representations treat the point cloud as an unordered set of discrete points with coordinates and optional attributes like colour or normals, preserving the raw, irregular nature of the data and making it ideal for sparse clouds by avoiding overhead from grid structures; processing often involves techniques like farthest point sampling (FPS) or k-nearest neighbors (KNN) for feature aggregation [36, 41, 42]. Octree-based representations organise points hierarchically by subdividing space into octants, creating a tree structure that adapts to varying densities through level-of-detail (LoD) refinement, improving efficiency in processing massive datasets [31, 43]. Voxel-based representations discretise space into a uniform grid where voxels indicate occupancy, enabling efficient convolution-based feature extraction for dense clouds, though they may introduce scaling inefficiencies in sparse scenarios [37]. Based on these representations, this process leverages redundancies in both geometric (spatial coordinates) and attribute data, such as colour, intensity, or surface normals, to facilitate the management of voluminous datasets while retaining critical quality for downstream applications, including immersive virtual reality, autonomous navigation, and digital preservation of cultural artefacts [44, 45].

PCC can be broadly classified into lossless and lossy compression categories.

Lossless PCC ensures that the reconstructed point cloud is identical to the original, with no loss of information, by eliminating statistical redundancies without discarding data [46, 47]. In contrast, lossy PCC allows for some information loss through techniques like quantisation, achieving significantly higher compression ratios while preserving acceptable quality for most practical applications [26, 40, 10]. Most research and evaluations in the field focus on lossy compression rather than pure lossless methods because point cloud data is typically massive in volume, and lossy approaches offer a superior RD trade-off, enabling efficient storage and transmission in bandwidth-constrained scenarios such as AR/VR and autonomous driving, where minor distortions are tolerable without compromising functionality [48, 44, 45].

Formally, in lossy learning-based PCC, a point cloud is denoted as a set of geometric coordinates $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^3 \mid i = 1, \ldots, N\}$, where N represents the number of points, often augmented with attributes $\mathbf{A} = \{\mathbf{a}_i \in \mathbb{R}^K \mid i = 1, \ldots, N\}$, where $\mathbf{a}_i$ encapsulates supplementary features like RGB values or reflectance properties that enhance realism and utility in analysis tasks, and K represents the dimensionality of the attributes. Conventional and learning-based PCC methods primarily aim to optimise an RD trade-off through minimisation of a loss function:

$$\mathcal{L} = \mathcal{D}((\mathbf{X}, \mathbf{A}), (\mathbf{X}', \mathbf{A}')) + \lambda \mathcal{R}, \tag{1.1}$$

where $\mathcal{D}((\mathbf{X}, \mathbf{A}), (\mathbf{X}', \mathbf{A}'))$ quantifies distortion between the input $(\mathbf{X}, \mathbf{A})$ and reconstruction $(\mathbf{X}', \mathbf{A}')$ (*e.g.*, using Chamfer Distance or Point-to-Plane Distance for geometry and Mean Squared Error for attributes [49, 50]), $\mathcal{R}$ represents the bitrate derived from the entropy of quantized latents, and λ balances these competing objectives.

In learning-based paradigms, the goal is to learn a neural model, $\mathcal{F}_\theta$, that encodes the point cloud into a compact bitstream and decodes it back:

$$\mathcal{F}_\theta : (\mathbf{X}, \mathbf{A}) \mapsto \mathcal{B} \mapsto (\mathbf{X}', \mathbf{A}'), \tag{1.2}$$

with $\mathcal{B}$ denoting the bitstream essential for complete reconstruction of geometry $\mathbf{X}'$ and attributes $\mathbf{A}'$.

1.2.2 Challenges of Point Cloud Compression

While traditional standards like G-PCC and V-PCC provide robust, scalable, and computationally efficient baselines, the transition towards learning-based architectures presents unique opportunities to push rate-distortion (RD) boundaries, albeit with associated challenges. This thesis specifically targets the following technical challenges in learning-based geometry compression [10, 3]:

A. Preserving Structural Geometric Details

Traditional PCC methods, such as octree-based [31] and voxel-based approaches [32], often rely on fixed hierarchical structures that struggle to adapt to irregular point distributions and density variations, leading to inefficiencies, increased bitrates, and artefacts in complex geometries [45, 44]. The reason is that they employ predefined partitioning schemes that fail to capture intricate local structures and global contextual relationships, especially for sparse or large point clouds. Learning-based methods, including Transformer architectures [36], variational autoencoders [51], and sparse tensor representations [37], offer improvements by learning data-driven features and achieving better RD performance compared to traditional approaches. However, they still have limitations, such as overlooking nuanced local-global feature interactions and introducing reconstruction artefacts in highly irregular datasets [11, 3].

B. Achieving High Accuracy at Low Bitrates

Traditional PCC methods suffer from rigid encoding strategies that lead to point cloud classification accuracy drops at low bitrates due to coarse quantisation and inadequate handling of high-dimensional data in decompressed point clouds [44]. These approaches are not sufficient as they prioritise simplicity over adaptability, resulting in poor rate-accuracy trade-offs and limited effectiveness in resource-constrained environments with diverse point cloud densities. In compressed-domain tasks such as compressed point cloud classification (CPCC), learning-based techniques provide superior performance by leveraging neural networks to preserve local features more effectively than traditional methods [2, 3]. Nevertheless, they remain limited, particularly with Transformer-based models that, while strong in capturing global contexts, often neglect density variations and intricate edge details, impacting overall accuracy on datasets with varied object classes [12].

C. Supporting Scalable Density-Adaptive Coding

While traditional point cloud compression standards like G-PCC [31], and V-PCC [32] support scalable decoding, their mechanisms operate primarily at the spatial or resolution granularity instead of the latent feature channel level. For example, the octree Level of Detail subdivisions and predictive tree (PredTree) coding within G-PCC facilitate spatial scalability across different resolution tiers [44]. Similarly, multiscale frameworks driven by deep learning, such as SparsePCGC [52] and Unicorn [53], achieve scalability across varying spatial resolutions through sequential refinement stages. Scalable point cloud compression (SPCC) within these neural network paradigms further addresses these structural issues by enabling hierarchical decoding and enhanced adaptability [54]. Nevertheless, current SPCC methodologies face substantial limitations. These include high computational com-

plexity, difficulties executing prioritization based on density across large datasets, and a reliance on multiple training stages, which ultimately prevents scalable decoding from a single training execution [3].

Crucially, achieving highly granular scalable geometry coding at the channel level utilizing a unified training paradigm remains an active area of research. In such systems, scalability emerges by dynamically pruning latent feature channels rather than relying on discrete spatial octrees or resolutions across multiple stages. Recent advancements driven by machine learning focus primarily on attribute compression, as demonstrated by Rudolph et al. [54]. To bridge this critical gap in geometry coding, there is a compelling need to draw inspiration from established scalable coding paradigms across channels and layers in 2D videos (e.g., H.264 SVC [30]). This adaptation enables efficient data delivery and dynamic bandwidth adaptability in constrained environments without claiming strict guarantees regarding low latency.

1.3 The Research Objectives

The overall goal of this research is to design learning-based methods that preserve geometric and structural details, maintain efficiency and accuracy at low bitrates, and enable scalable decoding for flexible, adaptive 3D data handling. By addressing the irregular structure, sparsity, and high data volume of point clouds, these methods aim to improve RD performance, reduce computational overhead, and enhance applicability across diverse domains such as autonomous navigation and immersive technologies [10, 3].

In detail, this research pursues three primary objectives:

1. Develop Transformer-based point cloud compression methods and test on ShapeNetCore.v2 [6], incorporating global attention with local patch mechanisms to preserve structural features and adaptive entropy modelling for op-

timising RD trade-offs, achieving superior geometry coding performance with enhanced Bjøntegaard Delta Rate (BD-Rate) performance compared to existing baselines [11, 38, 55].

2. Develop a local-attention-based edge-preserving CPCC method to achieve higher point cloud classification accuracy at lower bitrates and test on ModelNet40 [6], surpassing benchmark rate-accuracy trade-offs through an attention-based edge-centric feature extraction mechanism, and extend it to a global-attention-based CPCC for performance evaluation across varied compression levels [12, 2].

3. Develop a density-aware tail-drop method for SPCC to enable incremental decoding, drawing on the layered coding paradigm of H.264 SVC [30] by selectively discarding less critical features during a single training iteration, the proposed method can achieve scalable quality enhancements at increasing bitrates. It addresses irregular point cloud geometry by prioritising high-density regions indicative of intricate details [36, 54] and optimises RD performance compared with baseline methods on widely-used datasets like SemanticKITTI [5] and ShapeNet [6].

1.4 Contributions of This Thesis

To solve the research objectives, the proposed research makes contributions to the field of learning-based PCC, establishing new performance benchmarks:

1. **Transformer-Based Compression with Local Feature Preservation:** Despite advancements in learning-based PCC, gaps remain in adapting to irregular point distributions, where traditional methods like octree-based partitioning fail to capture local-global interactions, often leading to artefacts and fidelity [45, 44]. Inspired by hybrid architectures in computer vision and

graph neural networks, this thesis introduces a new Transformer-based autoencoder [11] that incorporates a Local Neighbour Aggregation module. This design enables the preservation of both global structures and intricate local features, which are essential for point cloud up-sampling and reconstruction. As a result, the proposed method achieves highly competitive RD performance compared to evaluated baselines [38, 56] on ShapeNetCore.v2 [6].

2. **Accurate Compressed-Domain Classification at Reduced Bitrates:** With the proliferation of LiDAR technologies, point clouds demand efficient compression, but traditional methods introduce artifacts degrading tasks like classification, prompting for direct inference on compressed data [45, 44]. A critical gap is the loss of structural details like edges during down-sampling, hampering semantic representation. Motivated by adaptive sampling methods [57, 58], this thesis proposes to leverage attention-based edge-sampling [12] to extract edge-centric features and details directly from the point cloud and prioritise and code the essential edge information into a bitstream. Compared with the recent baseline method, this approach delivers superior classification accuracy on ModelNet40 [6] at lower bits per point (BPP). Additionally, this thesis proposes the BD-Top-1 Accuracy metric [12] (analogous to BD-Rate, but calculating the average Top-1 classification accuracy [2] improvement across a range of bitrates) to quantify the trade-off between bitrate and accuracy, demonstrating robust accuracy improvements over previous methods at equivalent bitrates [2].

3. **Geometry Scalable Compression for Large Datasets:** The irregular nature of point clouds poses challenges for storage and transmission, with traditional PCC often generating monolithic bitstreams inefficient for real-world scenarios [3]. Achieving fine-grained, *channel-based* scalable geometry coding

utilising a single-training paradigm—where scalability is achieved by dynamically pruning latent feature channels rather than relying on discrete spatial octrees or multi-stage resolutions—remains an active area of research. While recent learning-based advancements focus primarily on attribute compression (e.g., Rudolph *et al.* [54]), this thesis bridges the gap in geometry coding by drawing inspiration from established channel/layer scalable coding paradigms in 2D videos, such as H.264 SVC [59, 30]. Specifically, this work proposes ScalaDAT, a new framework that enables SPCC through a single training stage, substantially reducing computational cost while enabling efficient data delivery and dynamic bandwidth adaptability in constrained environments. Central to ScalaDAT is a density-aware tail-drop mechanism that exploits point and structural density to prioritise regions of high structural significance, thereby enhancing efficiency by focusing on complex, dense areas and adaptively compressing sparser ones. The proposed method not only enables scalable point cloud coding but also achieves competitive coding efficiency compared to selected learning-based coding techniques [36, 31, 60], demonstrating consistent BD-Rate improvements for Point-to-Plane Peak Signal-to-Noise Ratio (PSNR-D2) on SemanticKITTI [5] and ShapeNet [6].

1.5 Thesis Organisation

This thesis is structured to systematically explore advancements in point cloud compression, focusing on learning-based methods that address key challenges in structural preservation, low-bitrate accuracy, and scalable decoding. The organisation is as follows:

- **Chapter 2**: A comprehensive literature review on point cloud compression techniques. It begins with point cloud representations and traditional methods, then transitions to learning-based approaches, categorised into point-

based, octree-based, and voxel-based methods. The chapter also discusses scalable point cloud compression, provides an overview of commonly used datasets, and outlines standard evaluation metrics.

- **Chapter 3**: Introduces a Transformer-based geometric point cloud compression method with local neighbour aggregation. This chapter highlights motivations and contributions, defines the problem, reviews relevant literature, details the proposed method [11] (including encoder, decoder, and loss function), describes experimental setups (datasets, implementation, and training details), evaluates performance comparisons, and concludes with key insights.

- **Chapter 4**: Focuses on compressed point cloud classification (CPCC) with point-based edge sampling. This chapter covers motivations and contributions, the problem statement, a literature review on point cloud downsampling and CPCC, the proposed method (overall architecture of CPCC-PES [12], attention-based edge sampling, neighbour attention aggregation, entropy bottleneck, decoder, and loss function), experimental results and discussion (datasets, implementation, performance comparisons, ablation studies, and edge-sampling visualisation), and a conclusion.

- **Chapter 5**: Proposes a scalable point cloud compression approach with density-aware tail-drop. This chapter discusses motivations and contributions, methodology (channel-importance based density-aware tail dropping and loss function), experiments (datasets and implementation, evaluation metrics, results, visualisation, and ablation studies), and a conclusion.

- **Chapter 6**: Summarises the overall conclusions of the thesis and outlines potential directions for future work. This is followed by the Bibliography.

Chapter 2

Related Works

Building upon the foundational concepts and challenges in point cloud compression (PCC) outlined in Chapter 1, this chapter provides a comprehensive literature review that contextualises the evolution of PCC techniques. It connects the background motivations—such as the need for efficient handling of irregularity and sparsity—to existing methodologies, categorising them by representations (*e.g.*, point-based for raw flexibility, octree-based for hierarchical scalability, voxel-based for CNN compatibility). Moreover, this thesis summarises significant published works in PCC, focusing on traditional and learning-based point cloud methods, datasets, evaluation metrics, limitations, cross-domain influences, the identified research gaps, and this thesis's contributions to address them. This literature review primarily emphasises the evolution of PCC methodologies, their limitations, and cross-domain adaptations (*e.g.*, from image/video compression), positioning this thesis's contributions in structure-preserving, edge-preserving, and scalable coding.

2.1 Overview of Point Cloud Compression

2.1.1 Point Cloud Representation

Point cloud representation is fundamental to efficient processing and compression, as it determines how 3D data is structured for algorithms to handle sparsity, irregularity, and scale. Common representations include point-based, octree-based, and voxel-based approaches, each offering distinct advantages in capturing geometry and attributes, which are extensively discussed in recent surveys on point cloud

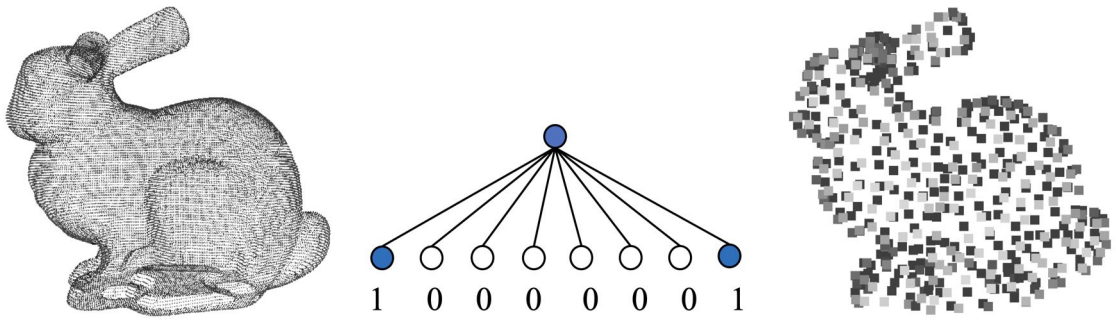
compression [10, 3].

Point-based representations treat the point cloud as an unordered set of discrete points, each defined by coordinates and optional attributes like colour or normals. This direct approach preserves the raw, irregular nature of the data, making it suitable for sparse point clouds where voxelisation might introduce unnecessary overhead. Processing typically involves farthest point sampling (FPS), k-nearest neighbours (KNN), multilayer perceptrons (MLPs), or graph-based networks to aggregate and extract features without imposing a grid structure [3, 41]. FPS iteratively selects maximally distant points for uniform coverage and redundancy reduction [61], offering robustness to point cloud global structures, outliers, and computational processing simplicity. KNN complements this by defining local neighbourhoods, enabling graph-based modelling of point relationships for capturing local geometric features, permutation invariance, and adaptability to varying densities [42]. Representative works include Density-preserving Deep Point Cloud Compression (D-PCC) [36], which uses MLPs and max-pooling for global and local feature extraction, enabling applications in compression and analysis.

Octree-based representations organise points hierarchically by recursively subdividing the 3D bounding box into eight octants, creating a tree where each node represents occupancy [62]. This method excels in handling varying densities, as it adaptively refines sparse regions while providing level-of-detail (LoD) structures for scalable compression [31]. To enhance dependency modelling, attention mechanisms are often incorporated, as they limit computations to correlated sibling and ancestor nodes, enabling large receptive fields and emphasising relevant contexts [43, 63]. Advantages include scalability for large-scale point clouds, improved efficiency in capturing long-range dependencies, and better performance in tasks like segmentation and detection. Representative works include Octsqueeze [40] and OctAttention [39], which extend receptive fields and capture large-scale contexts by emphasising cor-

related sibling and ancestor nodes, achieving significant bitrate reductions.

Voxel-based representations discretise the space into a uniform 3D grid of voxels, where each voxel indicates occupancy or stores aggregated point data. This regular structure facilitates convolutional neural networks (CNNs) for feature extraction, making it ideal for dense point clouds. The Minkowski Engine [64] is particularly effective for this, as it provides an auto-differentiation library for high-dimensional sparse tensors, supporting generalised convolutions that operate solely on non-empty voxels. Its advantages include significant reductions in memory usage and computation time, scalability to higher dimensions, and seamless integration of standard layers like pooling and normalisation for sparse data. To get higher performance, sparse convolutions [37, 48] are commonly employed, processing only occupied voxels to reduce computational overhead and address sparsity.



(a) Point-based (b) Octree-based (c) Voxel-based

Figure 2.1 : Conceptual representations of the three main PCC paradigms: (a) Point-based representation operating directly on raw spatial coordinates; (b) Octree-based representation employing hierarchical spatial subdivision, where leaf nodes are encoded as a binary sequence (‘1‘ for occupied, ‘0‘ for empty); and (c) Voxel-based representation utilising 3D uniform grids. Images adapted from Cao *et al.* [65] and [66].

To visually conceptualise these structural differences, Figure 2.1 illustrates the

fundamental paradigms of the three main data representations. While the point-based approach (Figure 2.1a) retains the explicit, continuous geometry of the target model, the voxel-based method (Figure 2.1c) approximates the shape using uniformly sized discrete cubic grids, inevitably introducing quantisation. Bridging the gap between structural accuracy and computational efficiency, the octree-based representation (Figure 2.1b) recursively subdivides the 3D space. Crucially, as shown at the bottom of the octree diagram, this hierarchical subdivision translates the 3D geometry into a flattened binary occupancy sequence (*e.g.*, 1 0 0 0 0 0 0 1). In this encoding scheme, a 1 signifies an occupied node (meaning the sub-space contains actual point data) that requires further processing, whereas a 0 denotes an empty space that can be efficiently bypassed or compactly compressed by entropy models.

Despite their advantages, these representations each have notable limitations. Point-based methods face scalability issues with large-scale point clouds, where quadratic computational complexity in graph construction or attention mechanisms leads to high memory and time demands [41]; they are also sensitive to noise and outliers from sensor data, potentially degrading feature extraction and requiring additional preprocessing steps like robust sampling, which can increase computational latency in bandwidth-constrained applications [3]. Meanwhile, octree-based approaches may lose fine-grained details in deeper leaf nodes due to fixed subdivision rules, introducing artifacts in highly irregular geometries [44], with substantial preprocessing time for tree construction in massive datasets and challenges in balancing tree depth with compression efficiency, which can lead to reduced rate-distortion efficiency in certain dynamic scenarios [3]. Furthermore, voxel-based methods suffer from quantisation artifacts that distort fine details during discretization and memory inefficiency for highly sparse clouds where most voxels remain empty, leading to unnecessary overhead [64, 37]; their cubic scaling with resolution further restricts applicability to high-resolution data, necessitating hybrid strategies to balance density

handling and computational costs [3, 10].

Essentially, all PCC works centre on these three representations, and although each has inherent limitations, ongoing research focuses on learning and integrating the distinct features of each representation, such as point-based flexibility for irregularity, octree-based hierarchical scalability, and voxel-based efficient CNNs processing for density, to mitigate these drawbacks, resolve specific challenges, and maximise their advantages through hybrid and adaptive approaches [10, 3, 25]. These hybrid approaches are further explored in subsequent sections on learning-based methods. This evolution underscores the importance of selecting and combining representations to optimise PCC performance across diverse applications, even in traditional PCC areas. Thus, in the next few sections, this thesis will discuss PCC methodologies, widely used datasets, and corresponding taxonomies.

2.1.2 Traditional Point Cloud Compression Methods

Traditional PCC has been developed and is widely used over the last few years. Regarding compression, this traditional coding task could be separated into point cloud geometry coding and point cloud attribute coding. Currently, the most representative works in this area are G-PCC [31] and V-PCC [32] proposed by MPEG. G-PCC employs octree representations and triangular surface coding to directly encode point clouds, making it particularly well-suited for static scenes. In contrast, V-PCC employs video codecs by projecting 3D point cloud data onto 2D planes, thereby taking advantage of established video compression techniques to compress the model.

Nevertheless, these standards have limitations that undermine their effectiveness in certain scenarios. For instance, G-PCC, while effective for static scenes, struggles in dynamic scenarios where it can face challenges in optimally exploiting temporal redundancies under certain configurations, resulting in lower compression ratios and

potential inefficiencies for time-varying point clouds [67]. In contrast, although V-PCC frequently attains high compression efficiency on temporally sequential data by exploiting temporal redundancy, this comes at the cost of additional overhead and high encoding complexity, as partitioning and projecting point clouds into video frames is computationally demanding and may introduce visual artefacts [68]. To address limitations in dynamic scenarios, recent standard extensions have introduced predictive tree (PredTree) coding to better exploit temporal redundancies in geometry across sequential frames [31]. As the primary focus of this thesis is on geometry compression, detailed discussions on attribute coding algorithms (e.g., RAHT) are omitted [69].

Overall, while traditional standards like G-PCC and V-PCC provide highly mature, fast, and robust baselines capable of handling varied point cloud formats, achieving extreme low-bitrate compression for complex irregular geometries can sometimes introduce quantisation artifacts, motivating the exploration of data-driven representations [11, 36].

2.1.3 Early Learning-based Point Cloud Compression Methods (Pre-2020)

In general, early learning-based PCC methods can be classified as methodologies based on CNNs [42] and Transformers [70], which researchers have adopted and integrated to enhance feature learning and reconstruction performance.

CNN-based PCC is a growing area of research as CNNs have shown high performance in processing and extracting features in point cloud compression tasks. Because standard 3D CNNs are designed for regular grids, most researchers voxelise point clouds prior to compression, which inherently introduces quantisation errors [26, 60]. Representative early methods include GeoCNNs [26], the PCGC series [60, 52], and autoencoder approaches [51]. However, standard 3D CNN-based

point cloud compression faces inherent limitations. Firstly, the voxelisation process introduces unavoidable quantisation artifacts and data loss that compromise fine geometric details. Secondly, applying dense convolutions to highly sparse point clouds is computationally inefficient, as resources are wasted on empty voxels unless mitigated by sparse convolutions [64]. Finally, standard convolutions possess a limited receptive field, which can restrict their ability to adaptively capture long-range global structures in irregular geometries [3, 10].

On the other hand, Transformer-based PCC methods leverage self-attention mechanisms [71] to address some of these CNNs' limitations by capturing relationships between the points and then applying quantisation and compression techniques to reduce the size of the data. This approach has shown promising results in reducing the storage and transmission costs of large point cloud datasets. Compared with CNN-based methods, Transformer-based methods have some advantages in processing point cloud features. Firstly, Transformers use attention mechanisms to capture and extract features, which enables them to process the irregularly unstructured point cloud. Secondly, Transformers have a larger receptive field. The self-attention mechanism used in Transformers allows each neuron to learn all input positions, resulting in a larger receptive field compared to CNNs. This larger receptive field, coupled with the ability to capture long-range dependencies, makes Transformers a powerful architecture for point cloud processing [11]. Some prominent proposals are D-PCC [36], Trans-PCC [72] and PCT-PCC [38]. However, even though these methods show superior performance in this area, Transformer-based methods have some limitations. Normally, this method requires high computational capability, posing challenges for large datasets [73]. To further improve the coding efficiencies of the PCC methods, more advanced techniques have been applied and integrated in this area recently.

2.2 Recent Learning-based Point Cloud Compression Methods (Post-2020)

To achieve higher RD performance, recent learning-based PCC methods (developed from 2020 onwards) utilise various advanced techniques. Building upon the foundational representations introduced in Section 2.1.1, these architectures are specifically designed to address the compression bottlenecks of point-based, octree-based, and voxel-based structures. A detailed comparison of these paradigms is summarised in Table 2.1.

2.2.1 Point-based Methods

Point-based methods can directly utilise the original characteristics of the data without information loss caused by voxelisation or other intermediate data transformations. For geometry compression, the point cloud is treated as an unordered set of points, and compression networks learn to extract geometric features from point coordinates and encode them into a compact latent representation. This paradigm mainly draws inspiration from PointNet++ [41] and its variants, which can aggregate local and global features from raw point coordinates.

Among the notable point-based works, Yan *et al.* [56] initiated this PointNet++ [41] direction. Their network uses an encoder to generate a latent code, followed by a uniform quantiser and entropy model before transmitting, and a decoder that reconstructs the points; this was the first codec to directly take raw point clouds as input rather than converting the points into voxels, outperforming G-PCC [31] on RD performance and demonstrating the potential of point-based PCC methods. Building on this, Wen *et al.* [74] introduced adaptive octree partitioning. This produces clustered subsets encoded by point-based networks, with this hybrid approach leveraging octree spatial local information while using point-

wise feature learning to yield better compression on large-scale scenes with varied densities. Similarly, Wiesmann *et al.* [75] introduced a new point-based scalable deconvolution operator that recovers original LiDAR point cloud information from small subsets of points using KPConv [76] layers, while D-PCC [36] focuses on retaining accurate local density distributions through sub-point convolutions and adaptive up-sampling to mitigate clustering artifacts, better capturing geometric features and obtain higher RD performance.

Furthermore, attention mechanisms enhance efficiency in point-based methods, for instance, as demonstrated by Gao *et al.* [77], who propose an attention-driven graph construction to select local points, thereby reducing computational costs while preserving essential geometric details—though this method may struggle with scalability in extremely large datasets due to graph complexity. Building on this foundation, feature aggregation is emphasised for improved quality, such as in the works of Zhang *et al.* [38] and Luo *et al.* [11], which capture long-range dependencies effectively and make attention-based methods outperform traditional solutions in handling sparse distributions across various applications; however, despite these strengths, Transformer-based models still have limitations on their practicality for semantic tasks. To further address the semantic preservation for downstream tasks, Xie *et al.* [78] introduce a method focusing on Regions of Interest (ROI) that guides geometry compression with a dual-branch framework consisting of a coarse encoding layer and an enhancement layer for residuals refined via ROI prediction networks, thereby enhancing fidelity in targeted regions, although it relies on accurate ROI detection, which could falter in noisy or ambiguous scenes. Extending this further to multi-modal integration, Cui *et al.* [79] present LAM3D, a large image-point-cloud alignment model that compresses point clouds into latent tri-planes using a Transformer and then aligns single-view image features via diffusion processes for high-fidelity 3D reconstruction; by fusing 2D and 3D modalities with attention-

driven inter-plane correlations, it achieves competitive mesh reconstruction while preserving point-wise details in sparse or irregular distributions. Finally, for optimised efficiency, Xu *et al.* [80] encode local geometry via KNN structures and introduce an implicit neural representation in the refinement layer for adaptable entropy models and arbitrary-density reconstruction.

These advanced point-based compression approaches demonstrate the versatility and effectiveness of end-to-end learning frameworks in extracting robust latent representations for point clouds with high reconstruction quality in both lossy and lossless scenarios, treating the data in its native format and excelling in scenarios with irregular or sparse distributions.

2.2.2 Octree-based Methods

While point-wise architectures enable end-to-end learning for point cloud analysis, octree-based methods offer a compelling alternative by hierarchically partitioning 3D space. This hierarchical organisation not only reduces computational complexity but also facilitates scalable compression, making octree-based approaches particularly suitable for handling large-scale, sparse point clouds such as those from LiDAR sensors. At their core, these methods encode point clouds into an octree structure, where each node represents a voxel subdivided into eight child nodes based on occupancy [31].

Learning-based extensions employ neural networks to predict occupancy probabilities for child nodes, conditioned on parent node information, sibling dependencies, and geometric features, enabling precise entropy modelling for arithmetic coding. The standardised G-PCC framework [31] by MPEG serves as a foundational benchmark, providing a generic pipeline for compressing diverse point clouds. Recent advancements build upon this to address quantisation artifacts and varying densities. For instance, G-PCC++ [81] enhances reconstruction quality through a

two-stage process: geometry recovery using kNN interpolation followed by neural-guided point selection, and attribute recolouring via Gaussian-weighted mapping with neural refinement. This approach effectively mitigates detail loss, achieving higher fidelity at comparable bitrates. Similarly, YOGA [82] introduces a hybrid compressor with a base layer that encodes down-scaled thumbnails using G-PCC for backward compatibility, augmented by an enhancement layer employing multiscale sparse convolutions, adaptive quantisation, and advanced entropy modelling. By integrating lossless base encoding with learned enhancements, YOGA adeptly handles heterogeneous densities and redundancies, demonstrating superior rate-distortion performance on diverse datasets.

Another pivotal work is OctSqueeze by Huang *et al.* [40], which pioneered a tree-structured conditional entropy model for LiDAR data compression. It leverages neural networks to predict child node occupancy distributions based on parent states and sibling contexts, marking a shift toward probabilistic modelling that exploits structural redundancies for significant bitrate reductions. However, its focus on parent and temporal contexts may undervalue sibling interdependencies, potentially limiting gains in highly irregular scenes. Addressing this, Que *et al.* 's VoxelContext-Net [29] employs a voxel-based entropy model to extract local features and compress non-leaf node symbols losslessly within the octree. By operating in two stages for static and dynamic compression, it achieves balanced efficiency. Further refining hierarchical representations, Fan *et al.* [83] utilise multiscale latent variables as side information to organise and compress data, reducing computation time while improving entropy estimation. These extensions collectively illustrate how expanding contextual scopes—spatial, temporal, and multiscale—scalably refines compression efficiency, though they highlight a trade-off: richer contexts boost performance but increase latency in real world scenarios. Other emerging trends in recent literature reinforce these developments while introducing new adaptations.

For example, density-adaptive octree methods [84] dynamically adjust voxel resolutions based on point distribution, further optimising for sparse large-scale scenes.

To capture even more comprehensive dependencies without prohibitive costs, attention mechanisms, originally from Vaswani *et al.* [70], have been adapted to octree structures. OctAttention [39] exemplifies this by modelling relationships among sibling and ancestor nodes via a tree-structured attention mechanism, enabling larger neighbourhood awareness and superior occupancy prediction. Building on this, OctFormer [43] restricts attention to local, non-overlapping blocks to handle large-scale data efficiently, processing multiple nodes in parallel to mitigate the sequential bottlenecks of traditional octree coding. Song *et al.* [85] advance this further with an efficient hierarchical entropy model that combines dual-Transformer frameworks and multi-group coding, extracting parallel contexts from diverse octree nodes to optimise both compression ratios and speed. However, these methods still face lingering challenges, such as neglecting fine-grained details and high computational demands. Wang *et al.* 's TopNet [86] addresses the challenge by synergising global and local feature extraction with sliding window attention and convolutional priors. This hybrid approach not only extends benchmarks but also suggests a pathway toward more robust, scalable models. These innovations underscore the evolving versatility of octree-based frameworks.

Overall, octree-based methods excel in generating robust latent representations through learned probability models, CNNs, and attention within a spatial hierarchy, consistently achieving highly competitive coding performance. While these approaches have dramatically improved compression efficiency, early probabilistic models often relied heavily on sequential autoregressive decoding, posing challenges for parallelisation. Fortunately, recent works like Song *et al.* [85] have actively addressed this bottleneck by introducing parallel multi-group coding for efficient context extraction.

2.2.3 Voxel-based Methods

Voxel-based approaches treat the point cloud as an occupancy grid format and apply 3D CNNs to compress it, effectively transforming the problem into a 3D occupancy compression [87]. This allows the effective utilisation of advanced deep image compression techniques such as end-to-end autoencoders.

Among the pioneering works in voxel-based methods, Quach *et al.* [26] proposed an analysis-synthesis codec paradigm using a CNN-based encoder to compress the binary voxel grid into a latent representation that is quantized and entropy-coded, followed by a CNN-based decoder to reconstruct voxel occupancy; however, it suffers from high temporal and spatial complexity due to dense convolutions on voxel grids, influencing subsequent efforts to mitigate computational overhead. To address this drawback, Guarda *et al.* [51] introduced strategies like spatial partitioning, which limits input size for networks and enables parallel block processing, while also applying block-based compression and training separate autoencoders for varying spatial regions or densities to adapt to local complexity, inspiring later adaptive and region-specific refinements. In the same year, Choy *et al.* [64] contributed sparse 3D convolutions, which focus operations on occupied voxels to reduce computation on empty spaces, a key influence on subsequent multiscale models by enabling efficient handling of sparsity. This was integrated by Wang *et al.* [60] into a multiscale autoencoder to dramatically reduce computation costs on empty voxels, though voxelisation still introduces precision loss for sparse clouds, prompting further multiscale advancements; they further advanced this with a multi-scale framework [52] that encodes a coarse voxel grid and scalably refines it level by level using prior outputs as context, merging octree progressivity with voxel CNNs modelling for parallel decoding within levels and sequential refinement across them, overcoming limitations in single-scale processing. Beyond autoencoders, Nguyen *et al.* [46] explored auto-regressive models for lossless compression with VoxelDNN, using masked con-

volutions to predict voxel occupancy probabilities fed into an arithmetic coder, but sequential dependencies increased complexity, leading to their own improvements via data augmentation and context extension [47] for better performance on sparse or noisy data; this sequential issue prompted a multiscale approach [88] for parallel probability estimation, reducing dependency and influencing later efficient entropy models.

Recent extensions have further refined voxel-based techniques, often addressing prior drawbacks like complexity and precision loss. Guarda *et al.* [1] developed JPEG Pleno Point Cloud as a representative architecture utilising Inception Residual Blocks [89] to capture point cloud voxel features effectively, building on block-based strategies to enhance adaptability. To further enhance the adaptability and handle varying densities more dynamically, Ruivo *et al.* [90] proposed adaptive super-resolution by integrating a dense module within learning-based PCC frameworks, addressing sparse and dense regions but potentially increasing overhead in uniform data. Moreover, inspired by attention mechanisms for irregular data, Liu *et al.* [91] employed KNN local self-attention in multiscale sparse tensors for capturing irregular point relationships, extending multiscale ideas to improve on fixed receptive fields in convolutions. Wang *et al.* [37] offered a sparse-tensor-centric view exploiting multiscale features to encode clusters with minimal redundancy and faster decoding, overcoming single-scale limitations though still reliant on voxelisation precision. Huang *et al.* [92] adopted a codebook-based hierarchical scheme for geometry quantisation to mitigate precision loss across densities, influencing quantisation-focused refinements but limited by codebook size in diverse data. Zheng *et al.* [93] fused point-based data with 2D view images via view-guided attention to reduce geometric redundancy and improve reconstructions, addressing voxelization artifacts but introducing dependency on view quality. Finally, Ye *et al.* [94] demonstrated LLM potential through cross-modality alignment techniques like clustering, k-tree, token

mapping, and LoRA to transform LLMs into point cloud compressors or generators, innovating beyond traditional neural approaches, though the computational demands of LLMs pose scalability challenges.

Unlike point-based methods, voxel methods excel in dense data but incur voxelisation overhead. These methods employ diverse data representations to process point clouds; they collectively showcase the power and potential for standardisation in learning-based PCC. These showcase potential for standardisation. The next section elucidates differences by data type.

2.2.4 Comparisons among These Categories

To better elaborate on the analysis and conclusion of various learning-based PCC methods based on data types, we present a comprehensive comparison in Table 2.1. This table highlights advantages, disadvantages, and applications, allowing a nuanced understanding of trade-offs. For example, point-based methods excel in high-fidelity applications due to direct processing without quantisation losses, but their higher computational costs (*e.g.*, in works like He *et al.* [36]) limit scalability for large datasets. In contrast, octree-based approaches, such as Fu *et al.* [39], offer efficient handling of LiDAR data through hierarchical scalability, though they face challenges with dynamic point clouds and attribute compression.

Furthermore, voxel-based methods provide compatibility with CNNs for dense clouds but are inefficient for sparse data, as evidenced by overhead in sparse convolutions [37]. Table 2.1 underscores these differences, showing how hybrid adaptations (*e.g.*, combining point-based flexibility with octree hierarchy) can mitigate drawbacks, drawing from cross-domain influences like image compression standards.

Table 2.1 : Comparison of Learning-Based Point Cloud Compression Methods

Advantages	**Disadvantages**	**Applications and Representative Works**
Point-based		
1. Direct processing of point geometry and attributes. 2. Preserves fine geometric details with high precision. 3. No information loss from quantisation. 4. Flexible representation achieving high reconstruction quality.	1. Higher computational cost from individual point cloud processing. 2. Lower efficiency due to lack of spatial structure. 3. Difficulty processing large-scale point clouds. 4. Higher entropy reduces compression efficiency. 5. High computational latency without hardware-specific optimisation.	1. High-fidelity applications like heritage storage and scanning. 2. Research-focused learning-based compression. 3. Works: [56, 74, 75, 36, 93, 80, 11, 79, 78, 95, 96].

Continued on next page

Table 2.1 Continued from previous page

Advantages	**Disadvantages**	**Applications and Representative Works**
Octree-based		
1. Fast coding with adaptive multi-resolution. 2. Efficient encoding of irregular data with scalable compression ratios. 3. High space utilisation by partitioning only non-empty regions. 4. Hierarchical spatial information with strong RD performance.	1. Information loss from octree subdivision in dense point clouds. 2. High computational cost for structured octree processing. 3. Intensive recursive subdivision and tree traversal. 4. Poor handling of dynamic point clouds. 5. Less effective for attribute compression.	1. Static scenes and LiDAR data. 2. Map storage and 3D scan compression. 3. Works: [40, 97, 29, 39, 83, 43, 85, 86, 84, 98].

Continued on next page

Table 2.1 Continued from previous page

Advantages	**Disadvantages**	**Applications and Representative Works**
Voxel-based		
1. Regular grid structure enables efficient spatial partitioning. 2. Storage reduction by representing point groups as single voxel values. 3. Compatible with 3D CNNs and image/video codecs. 4. Simple and fast encoding/decoding.	1. Loss of fine details, especially in dense regions or with larger voxel sizes. 2. High memory usage from redundant empty space storage. 3. Extra voxelisation overhead compared to Point-based methods. 4. Inefficient for sparse point clouds.	1. Dense point clouds requiring 2D codec integration. 2. Telepresence and AR streaming. 3. Works: [26, 29, 60, 1, 51, 93, 52, 90, 91, 37, 85, 46, 47, 88, 92, 94].

2.2.5 Learning-based Point Cloud Compression Limitations

Based on above comprehensive analysis and categorisation, learning-based PCC methods have leveraged neural networks to achieve superior RD performance compared to traditional methods. However, they still encounter several limitations that present specific research opportunities to further optimise how networks handle the unique characteristics of point cloud data, such as irregularity, sparsity, and high dimensionality. These challenges are evident in structural detail preservation, accuracy at low bitrates, scalability for scalable coding, and additional issues like multimodal integration [38, 2, 36]. Addressing these limitations is crucial for enhancing the applicability of learning-based PCC in real-world scenarios, as highlighted in

recent surveys and empirical studies [3, 10].

a. Structural detail preservation:

Maintaining fine-grained features, such as edges, textures, and local geometries, in irregular and sparse point cloud data remains a significant challenge for learning-based PCC. Many early autoencoder-based models that prioritise global feature extraction tend to over-smooth intricate details during the encoding-decoding process, especially in lossy compression scenarios where quantization introduces artifacts [36, 99]. For instance, variational autoencoders and sparse voxel networks may effectively reduce redundancy but may not adequately preserve topological embeddings and geometric relationships in sparse regions, leading to fidelity losses in reconstructed clouds [100]. Additionally, sensor-induced noise and outliers exacerbate these issues, as current models lack robust mechanisms for noise filtering or outlier rejection during compression, resulting in degraded accuracy for downstream analytical tasks or high-fidelity 3D asset retrieval, where precise structural preservation is essential [101, 102]. Recent works emphasise that while density-preserving techniques mitigate some over-smoothing by adaptively handling local densities, they still struggle with highly irregular data distributions, underscoring the need for more adaptive feature preservation strategies [103, 104].

b. Accuracy under constrained bitrates:

Learning-based PCC methods face inherent trade-offs between compression ratio and reconstruction quality, quantified by metrics such as PSNR. At low bitrates, distortion becomes pronounced due to aggressive quantisation and latent space compression, which discards critical information and leads to significant accuracy degradation in downstream tasks like semantic segmentation or object de-

tection [2, 26]. For example, Transformer-based compressors excel in capturing long-range dependencies but often neglect fine-scale features, resulting in poor rate-accuracy balances in bandwidth-limited environments [12, 38]. This is particularly challenging in scenarios involving diverse datasets, where models trained on high-density clouds underperform on sparse inputs, amplifying errors in classification or detection pipelines [48]. Empirical evaluations show that while diffusion models offer promising low-bitrate performance, they still suffer from cumulative distortions that impact task-specific utility, highlighting the necessity for bitrate-adaptive optimizations to maintain acceptable accuracy without excessive computational cost [73].

c. Scalability coding:

Handling large-scale point clouds, such as those with billions of points from urban LiDAR scans, poses scalability challenges for learning-based PCC, primarily due to high computational demands in encoding and decoding phases. Current methods often require substantial memory and processing resources, limiting their efficiency for bandwidth-constrained streaming on mobile devices or in streaming contexts [54, 105]. While multiscale learning-based frameworks (*e.g.*, SparsePCGC [52]) have successfully introduced foundational scalable capabilities, achieving fine-grained, density-aware scalable geometry compression utilising a single-training paradigm remains an active area of research, leading to inefficiencies in dynamic bandwidth scenarios where incremental refinement is needed for low-latency rendering in AR/VR or telepresence [106]. While traditional standards like G-PCC inherently support scalable decoding via octree hierarchies, many early learning-based single-rate encoders necessitate full bitstream decoding, which is impractical for massive datasets, and existing scalable approaches face challenges in hierarchical feature selection across varying scales [107, 108]. These limitations underscore the demand for resolution-scalable frameworks that support adaptive

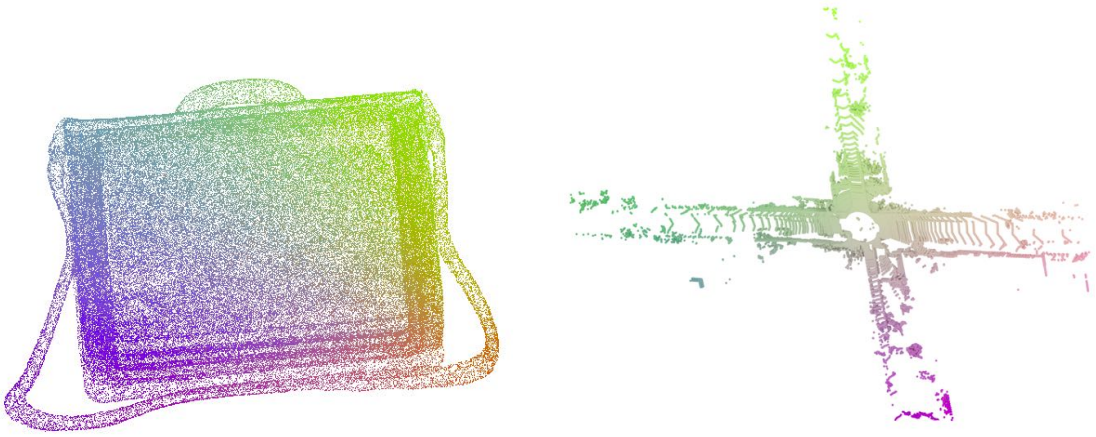
decoding, ensuring efficient handling of large volumes while facilitating scalable enhancements for resource-constrained environments [109].

d. Additional challenges:

Integrating attributes (*e.g.*, colour, intensity) with geometry without amplifying reconstruction errors is another hurdle, as separate encoding often results in misalignment and compounded distortions in multi-modal compression involving geometry, colour, and semantics [40, 46]. Besides, emerging challenges include managing multi-modal data fusion, where semantic attributes require specialised handling to avoid error propagation, and addressing generalisation across diverse acquisition modalities like LiDAR versus photogrammetry [3, 10]. The dependency on large training datasets and the lack of robustness to domain shifts limit practical adoption is another challenge, as models may overfit to specific distributions, emphasizing the need for more generalisable, data-efficient architectures in future research [25].

2.3 Commonly Used Datasets

In both classical and learning-based point cloud compression (PCC), datasets serve as foundational benchmarks enabling standardised evaluation of compression efficiency, reconstruction fidelity, and scalability across diverse applications. To test the compression performance of different methodologies, researchers typically utilise various benchmark datasets selected based on their scale, diversity, and structural complexity, which are critical for evaluating PCC methods under real-world conditions. As visualised in Figure 2.2, point cloud characteristics vary dramatically depending on their acquisition method and application domain.



(a) Dense Synthetic Object (ShapeNet [6])

(b) Sparse Real-world LiDAR Scan (SemanticKITTI [5])

Figure 2.2 : Visual comparison of point cloud characteristics across diverse application domains. (a) A dense synthetic object requiring high-fidelity surface preservation. (b) A sparse real-world LiDAR scan. Rather than targeting autonomous driving, this thesis primarily focuses on high-fidelity 3D asset archival (a) and bandwidth-constrained scalable streaming (b).

For instance, large-scale LiDAR datasets like SemanticKITTI [5] (illustrated in Figure 2.2b), with approximately 4.5 billion points, are ideal for testing sparsity and scalability in learning-based approaches. They simulate environments where methods must efficiently handle varying densities and bandwidth-constrained transmission. In contrast, ShapeNet's [6] dense, object-centric models (*e.g.*, with ∼55,000 points per shape), as shown in Figure 2.2a, are suited for assessing fine-grained surface reconstruction. While LiDAR datasets like SemanticKITTI are crucial for evaluating scalable coding, this thesis heavily leverages dense synthetic datasets like ShapeNet to rigorously evaluate the preservation of high-fidelity local structures. Table 2.2 summarises these datasets grouped by application domains, highlighting features such as point count, sparsity, and usage in tasks like segmentation or

detection.

As shown in Table 2.2, dynamic sequences (*e.g.*, SemanticKITTI [5] with ∼120k points/frame) emphasise temporal redundancy, making them dependent on octree-based methods for hierarchical processing, while indoor scenes like ScanNet [24] support voxel approaches due to higher density. This diversity enables deeper searches for multi-faceted reasoning, as seen in chronological analyses for comprehensive answers [3].

Table 2.2 : Widely Used Point Cloud Datasets Grouped by Application Domains

Dataset	**Features**
Synthetic CAD Models	
ModelNet40 [6]	Contains ∼12,311 synthetic CAD models across 40 categories; with ∼1–2k points per model (sampled); Widely used for 3D object classification and shape analysis
ShapeNet [6]	Includes ∼51,300 richly annotated synthetic CAD models spanning 55 categories; with ∼55,000 points per model; Models can be sampled to generate point clouds
Indoor Scenes	
ScanNet [24]	Comprises 1,513 indoor RGB-D reconstructed scenes; each with ∼3.3M points; Mainly used for scene understanding and segmentation
S3DIS [110]	Features 271 rooms from indoor LiDAR scans of office environments; with ∼1M points per room; Annotated with semantic labels for segmentation; High point density in complex areas
Autonomous Driving Scenarios (Sparse LiDAR)	

Continued on next page

Table 2.2 Continued from previous page

Dataset	Features
KITTI [5]	Includes ∼43,000 frames of automotive LiDAR scans from urban/suburban driving scenarios; with ∼120k points per frame; High sparsity with points concentrated on vehicles; Widely used for autonomous driving tasks
nuScenes [111]	Contains 1,000 scenes with ∼390k LiDAR sweeps; each with ∼34k points; Multi-modal autonomous driving dataset; Normally used for 3D detection and tracking
Waymo Open [112]	Comprises 1,150 scenes from automotive LiDAR; with ∼177k points per frame; Captured with multiple LiDAR sensors resulting in dense point clouds; Annotated as detection and mapping
Volumetric Video (Dense Dynamic Sequences)	
MVUB [16]	Includes 5 dynamic point cloud sequences of human upper bodies captured via depth sensors, with ∼300k points per frame; Used in volumetric video research
8i VFB [113]	Features 4–5 dynamic point cloud sequences of human performances with color; with ∼1M points per frame; Widely used in PCC research

From Table 2.2, *Classical methods*, such as G-PCC [31], excel on sparse LiDAR data (*e.g.*, SemanticKITTI [5]), where hierarchical structures like octrees effectively handle irregularity and temporal redundancy in dynamic sequences. For instance, datasets from autonomous driving domains, including KITTI [5] and nuScenes [111], feature high sparsity and real-world noise from sensor artifacts, testing classical codecs' robustness to environmental variations without heavy reliance on learned features. Similarly, dynamic human-capture sequences like MVUB [16] and 8i

VFB [113] prioritise temporal compression in classical approaches, enabling efficient encoding for volumetric video applications. However, classical methods may underperform on dense synthetic data due to fixed partitioning schemes that introduce artifacts in intricate geometries.

Learning-based approaches, on the other hand, leverage dense synthetic sets (*(e.g.*, ShapeNet [6]) for feature learning, where models like Transformers [38] or autoencoders [11] train on intricate geometries to capture local-global dependencies. For example, ShapeNet's dense, surface-rich models suit voxel-based learning methods [60, 52, 11, 96] for multi-scale feature extraction, while ModelNet40 [6] supports point-based techniques [36, 38] for classification-oriented compression. Indoor scenes like ScanNet [24] and S3DIS [110] bridge paradigms, facilitating learning-based segmentation tasks with semantic annotations, whereas classical methods use them for voxel partitioning efficiency. Additionally, autonomous driving datasets enable generative tasks in learning-based PCC, such as LiDAR scene synthesis [114], contrasting classical focus on raw compression.

Notably, ShapeNet [6] and SemanticKITTI [5] garner significant attention due to their extensive categories, high point counts, structural complexity, and large scales relative to other benchmarks. For example, ShapeNetCore.v2 [6] includes $\sim$ 51,300 models across 55 categories, with dense sampling (*e.g.*, 2,048 points per object) capturing intricate surfaces and diverse shapes like furniture and vehicles—surpassing smaller sets such as ModelNet40 ($\sim$12,311 models in 40 categories). Likewise, SemanticKITTI features 43,552 LiDAR scans from 22 sequences, yielding $\sim$4.5 billion sparse, semantically annotated points in $\sim$85 GB of raw data, offering greater real-world sparsity and scale than indoor datasets like ScanNet [24], which contains 1,513 scenes from over 2.5 million frames with millions of points per scene focused on static, dense environments. Consequently, SemanticKITTI's sparsity favours octree-based analysis, as explored in [36, 40, 29, 39, 43, 115, 28, 85, 63, 37]. In con-

trast, ShapeNet's dense, surface-rich clouds suit voxel-based approaches, as explored in [60, 52, 11, 96].

In the field of learning-based PCC, direct comparisons across different methodologies using benchmark datasets like ShapeNet and SemanticKITTI are often challenging due to the inherent dependencies on preprocessing steps tailored to specific network architectures. For instance, Voxel-based methods [26, 29, 60, 1, 51, 93, 52, 90, 91, 37, 85, 46, 47, 88, 92, 94] require voxelisation to create occupancy grids for convolutional processing, Octree-based approaches [40, 97, 29, 39, 83, 43, 85, 86, 84, 98] necessitate hierarchical partitioning into tree structures for occupancy prediction, and Point-based techniques [40, 97, 29, 39, 83, 43, 85, 86, 84, 98] operate directly on raw, unordered points without intermediate transformations. This preprocessing variability introduces inconsistencies in input representations, making unified rate-distortion evaluations unreliable, as the compression artifacts, feature extraction efficiency, and reconstruction fidelity are influenced by these foundational differences. Consequently, fair assessments typically occur within the same category, such as Octree-based methods [39, 43, 29, 40, 63]. Furthermore, disparities arise from divergent focuses on compression aspects, such as geometry-only versus joint geometry-attribute coding, lossy versus lossless modes, or scalability to dynamic sequences, which are not uniformly addressed across studies. Other challenges also include the selective reporting of results on subsets or other inconsistent datasets [92, 91, 93, 38, 11], all of which hinder standardized, cross-paradigm benchmarking despite shared benchmarks like ShapeNet and SemanticKITTI.

When categorizing methods into point-based, voxel-based, and octree-based paradigms and evaluating solely on geometry compression performance, notable leaders emerge. In the point-based category, D-PCC [36] stands out, achieving up to 15% better RD performance on sparse data for its density-preserving convolutions and adaptive up-sampling, effectively handling irregular distributions. For voxel-

based approaches, Wang *et al.* [37] excels particularly on SemanticKITTI through its sparse tensor-centric multiscale feature exploitation, while Guarda *et al.* [1], proposed as part of the JPEG Pleno standard, serves as a benchmark with strong performance on its designated datasets, though direct head-to-head comparisons with methods like Wang *et al.* on identical benchmarks remain limited. Octree-based methods exhibit intense competition, with Wang *et al.* [86] currently leading through its Transformer-efficient occupancy prediction that balances local and global dependencies via hybrid convolutional-attentional mechanisms and achieves a better RD performance.

Challenges in dataset usage include preprocessing variability (*e.g.*, voxelization for classical vs. raw points for learning-based), which affects fair comparisons, and domain shifts (*e.g.*, sparse outdoor vs. dense indoor) [80]. Besides, emerging trends involve multi-modal datasets (*e.g.*, nuScenes with sensor fusion) for joint geometry-attribute coding. These datasets influence our work by enabling validation of scalable coding on sparse (SemanticKITTI) and dense (ShapeNet) data, highlighting density-aware adaptations. For instance, SemanticKITTI's sparsity informs our tail-drop prioritisation of high-density regions (*e.g.*, vehicles), reducing artifacts in low-bitrate scalable decoding, while ShapeNet's geometric diversity validates our method's scalability across object categories. Drawing from cross-domain influences like LiDAR scene generation [114], this setup bridges classical efficiency with learning-based fidelity, addressing gaps in scalable geometry for bandwidth-constrained streaming. To further explain how to measure the PCC methods on these benchmark datasets, the next section will discuss methods of evaluation.

2.4 Evaluation Methodologies

Evaluation of PCC methods bridges classical and learning-based PCC, assessing RD trade-offs across paradigms. They rely on objective metrics for their re-

producibility and efficiency, supplemented by task-specific measures where applicable. Key metrics include bits per point (Bpp) for bitrate efficiency, Chamfer Distance (CD) for geometric similarity, Peak Signal-to-Noise Ratio Distance1 (PSNR-D1) and Distance2 (PSNR-D2) for reconstruction quality, and Bjøntegaard Delta Rate (BD-Rate) for overall compression performance [49, 116]. For classification tasks in Compressed Point Cloud Classification (CPCC), this thesis leverages the proposed BD-Top-1 Accuracy [12] to assess trade-offs between bitrate and performance [117, 41, 2], which is further explained in Sec. 4.4.2. In Scalable Point Cloud Compression (SPCC), lacking standardized benchmarks, this thesis adopts standard PCC metrics aligned with prior benchmark works [54, 105]. These metrics enable direct comparisons across this thesis, quantifying the proposed methods' improvements in RD efficiency, geometric fidelity, and adaptability for scalable coding.

In detail, widely used evaluation methodologies can be defined as follows:

a) Bits per Point

Bpp quantifies the average bitrate per point in a compressed point cloud, providing a direct measure of compression efficiency [116]. For a compressed file of size R bits containing N points, Bpp is defined as:

$$\text{Bpp} = \frac{R}{N}. \tag{2.1}$$

Lower Bpp indicates higher compression ratios, often separated into geometry and attribute components for detailed analysis [49]. In RD plots, Bpp serves as the x-axis against distortion metrics like PSNR or CD. In classical methods like G-PCC [31], BPP is primarily used to evaluate storage and transmission efficiency on large-scale datasets, such as SemanticKITTI [5], where lower values (*(e.g.*, <1 BPP) indicate effective redundancy reduction in sparse LiDAR scans. In contrast, learning-based methods, such as Transformer architectures [38], leverage BPP along-

side adaptive entropy models to optimise for variable-rate scenarios, enabling finer control in bandwidth-constrained applications like AR/VR streaming. This thesis uses Bpp to evaluate the bitrate savings of different methods, demonstrating superior efficiency over baselines in bandwidth-constrained scenarios.

b) Chamfer Distance

CD measures geometric disparity between point sets, serving as both an evaluation metric and training loss in PCC [118, 119]. For the ground truth point cloud $A = a_i \in \mathbb{R}^3$ and the decoded point cloud $B = b_j \in \mathbb{R}^3$, where $|A|$ and $|B|$ denote the cardinalities (number of points) of A and B respectively, CD is defined as:

$$d_{\mathrm{CD}}(A, B) \;=\; \frac{1}{|A|}\sum_{a\in A}\min_{b\in B}\|a-b\|_2^2 \;+\; \frac{1}{|B|}\sum_{b\in B}\min_{a\in A}\|a-b\|_2^2. \tag{2.2}$$

Lower CD values signify closer shape alignment, penalising unmatched points effectively. Classical approaches, including octree-based G-PCC [31], use CD to assess structural integrity in voxelised representations, particularly on indoor datasets like ScanNet [24], where it highlights artifacts from fixed partitioning (*e.g.*, CD values greater than 0.01 indicate over-smoothing in sparse regions). Learning-based methods, such as autoencoders [11, 38], employ CD as both an evaluation metric and training loss, improving local feature preservation in dense synthetic data like ShapeNet [6]. CD is employed to assess the proposed methods' ability to preserve structural details, validating enhanced fidelity in reconstructions compared to baseline approaches.

c) Peak Signal-to-Noise Ratio Distance1

PSNR-D1, as a standard in MPEG assessments for reconstruction quality [120], evaluates point-to-point geometric error. Assuming the ground truth point cloud is

$A = a_i \in \mathbb{R}^3$ and the decoded point cloud is $B = b_j \in \mathbb{R}^3$, where $|A|$ and $|B|$ denote the cardinalities (number of points) of A and B respectively, the directional mean squared error (MSE) from A to B is:

$$e^{D1}_{A,B} \;=\; \frac{1}{|A|}\sum_{a\in A}\min_{b\in B}\|a-b\|_2^2, \tag{2.3}$$

with overall MSE as $\text{MSE}_{D1} = \max(e^{D1}A, B, e^{D1}B, A)$. PSNR-D1 is:

$$\text{PSNR-D1} \;=\; 10\log_{10}\left(\frac{P^2}{\text{MSE}_{D1}}\right), \tag{2.4}$$

where P is the peak coordinate value. Higher PSNR-D1 values indicate reduced distortion, making it ideal for benchmarking geometric accuracy. In classical PCC, such as V-PCC [32], PSNR-D1 benchmarks accuracy in projected 2D encodings on dynamic sequences like MVUB [16], focusing on Euclidean fidelity for navigation tasks. Learning-based paradigms, including density-preserving models [36], extend PSNR-D1 to quantify latent space optimisations, achieving higher values at low bitrates on autonomous driving data like KITTI [5], where it validates robustness to noise. Usually, PSNR-D1 is leveraged to benchmark the geometric accuracy of methods, highlighting advantages in high-fidelity compression.

d) Peak Signal-to-Noise Ratio Distance2

PSNR-D2 assesses point-to-plane error along surface normals, emphasising perceptual quality and surface smoothness. For the ground truth point cloud $A = a_i \in \mathbb{R}^3$ and the decoded point cloud $B = b_j \in \mathbb{R}^3$, where $|A|$ and $|B|$ denote the cardinalities (number of points) of A and B respectively, the directional MSE from A to B is:

$$e^{D2}_{A,B} \;=\; \frac{1}{|A|}\sum_{a\in A}\left[(a-b)\cdot \mathbf{n}_b\right]^2, \tag{2.5}$$

with $\text{MSE}_{D2} = \max(e^{D2}A, B, e^{D2}B, A)$. PSNR-D2 is:

$$\text{PSNR-D2} \;=\; 10\log_{10}\left(\frac{P^2}{\text{MSE}_{D2}}\right). \tag{2.6}$$

Higher values reflect better normal-aligned reconstructions. Classical methods like G-PCC [31] apply PSNR-D2 to hierarchical structures on cultural heritage scans (*e.g.*, Semantic3D [23]), ensuring normal-aligned reconstructions. In this thesis's evaluations, PSNR-D2 quantifies perceptual improvements, persuasively demonstrating the methods' superiority in maintaining surface integrity. In learning-based contexts, PSNR-D2 is crucial for AR/VR applications, where voxel-based networks [37] use it to mitigate quantisation artifacts in dense indoor scenes like S3DIS [110]. This thesis leverages this to quantify perceptual improvements, persuasively demonstrating the methods' superiority in maintaining surface integrity.

e) Bjøntegaard Delta Rate

BBjøntegaard delta rate (BD-Rate) compares RD curves, quantifying bitrate savings at equivalent quality levels via cubic spline interpolation [121]. Let $R_1(D)$ and $R_2(D)$ be the RD functions (rate as a function of distortion) for the reference and proposed methods, respectively, interpolated over a common distortion range $[D_{\min}, D_{\max}]$. BD-Rate is computed as:

$$\text{BD-Rate} \;=\; \frac{1}{D_{\max} - D_{\min}} \int_{D_{\min}}^{D_{\max}} \left(R_2(D) - R_1(D)\right) dD. \tag{2.7}$$

Negative values indicate efficiency gains. BD-Rate is applied to evaluate Learning-based methods that adapt BD-Rate for neural codecs [11, 36], enabling comparisons on ShapeNet [6] where negative values (*e.g.*, -20%) indicate superior scalability—for example, Transformer models achieve -30% on dense objects. BD-Rate is applied in this thesis to evaluate overall performance, rigorously showing the proposed methods' bitrate reductions while preserving quality across distortion ranges.

2.5 Cross-Domain Influences

To achieve higher rate-distortion (RD) performance, as measured by the aforementioned evaluation metrics across diverse datasets, researchers have derived significant benefits from cross-domain inspirations, particularly from advancements in image and video compression [59], Transformer-based architectures for sequence and spatial modelling [71], and density-aware techniques from computer graphics and vision [36]. By adapting these strategies, PCC achieves enhanced RD performance, better handling of irregular data, and improved efficiency for bandwidth-constrained streaming [44, 10]. These influences collectively address key PCC limitations, such as structural detail preservation through global-local attention mechanisms that improve fidelity in irregular data by dynamically weighing dependencies [122, 117, 38, 123], accuracy under constrained bitrates via edge-centric strategies that retain essential geometry through down-sampling for superior rate-accuracy trade-offs [124, 125, 87, 126, 3], and scalable coding from 2D/video domains enabling low-latency transmission [127, 108, 109, 10, 54, 105]. Overall, these cross-domain influences alleviate PCC challenges and pave the way for robust, efficient compression paradigms.

Transformer architectures, extended from natural language processing to spatial domains, provide powerful tools for PCC by capturing long-range dependencies and hierarchical features. Global attention models overall scene context, local attention focuses on neighborhoods, and hybrid variants balance both for robust representations [122, 123, 128, 129, 38, 72]. These surpass traditional PCC by enabling adaptive feature interactions, with cross-domain adaptations from image segmentation or video analysis enhancing multi-scale attention and attribute integration without error amplification [130, 117, 11].

Inspirations from image and video compression include edge-aware coding, which

preserves structural features by prioritising boundaries and high-frequency details during quantisation and entropy coding [99, 29]. Adapted to PCC, these techniques combat over-smoothing in lossy methods and handle sensor noise or outliers, using edge detection via parametric inference or capsule networks to maintain fine-grained features in sparse data [124, 131, 101, 125].

Additionally, scalable coding from video standards like H.264 Scalable Video Coding (SVC) extension [30] enables layered transmission for partial reconstruction, inspiring the solution for PCC's challenges with large-scale clouds. Scalable methods, such as deep codecs using trit-planes or variance-aware masking, support incremental refinement under bandwidth constraints, facilitating adaptive streaming in AR/VR point cloud applications [108, 109, 132, 54, 105]. Density-aware techniques from graphics and vision, such as LOD rendering and adaptive sampling [103, 133], prioritise regions of varying density to address irregularity and multi-modal challenges. Adaptive sampling dynamically adjusts resolution based on complexity, ensuring efficient encoding for downstream tasks like segmentation or detection [134, 126, 87].

2.5.1 Progressive Coding vs Scalable Coding

PCC continuously adapts techniques from traditional 2D image and video standards to address bandwidth constraints, clarifying the terminology for bitstream delivery becomes critical. In the broader deep learning literature, the terms *progressive* and *scalable* are sometimes used interchangeably to describe systems that incrementally improve reconstruction quality. However, within strict multimedia standardisation contexts (*e.g.*, JPEG [49] and MPEG [135]), they represent fundamentally distinct paradigms regarding bitstream structure and decodability.

Progressive Coding: Progressive coding refers to a continuous bitstream where the transmission can be interrupted or truncated at any arbitrary point, yet still

yield a valid, decodable output. A foundational cross-domain example is the JPEG 2000 still image compression standard [136], which utilises embedded block coding to ensure the bitstream is optimally ordered by importance, allowing progressive refinement by pixel accuracy or resolution. In the context of PCC, a strictly progressive bitstream transmits geometric information sequentially (*e.g.*, point by point). If a network link is suddenly lost, the decoder can still successfully reconstruct and plot all the points received prior to the interruption. Crucially, every additional few bits grabbed by the decoder immediately contribute to decoding correspondingly more points or enhancing the resolution. Therefore, the codec offers highly granular, continuous quality improvements where decoding can stop at any time without rendering the previously received partial bitstream useless (*e.g.*, successfully decoding $X\%$ of the points).

Scalable Coding: Conversely, scalable coding structures the bitstream into discrete, hierarchical layers, typically comprising a base layer and one or multiple enhancement layers. This paradigm is prominently exemplified by the Scalable Video Coding (SVC) extension of the H.264/AVC standard [137], which provides spatial, temporal, and quality scalability. In a scalable PCC codec, the data arrives in set layers (*e.g.*, discrete levels of detail or hierarchical octree layers). Upon receiving the entire base layer, a low-resolution version of the complete point cloud can be decoded. To improve this initial quality, the decoder must receive the *entire* subsequent enhancement layer, along with all preceding layers. Unlike progressive coding, scalable coding cannot be meaningfully interrupted at any arbitrary time to yield partial improvements. In the event that the decoder successfully acquires the initial layer but only partially receives the subsequent layer, the truncated data fails to yield any reconstructive quality gain; consequently, the decoder must revert to the quality of the last fully received layer.

Terminology Alignment in This Thesis: Recognising this strict distinction

is vital for accurately defining the methodologies proposed in this thesis. While both paradigms address dynamic network bandwidths, the model proposed in subsequent chapters (*e.g.*, 5) align strictly with the definition of scalable coding. It processes and transmits geometric features in discrete, density-aware hierarchical layers, ensuring structured quality enhancements only when full layers are received, rather than providing continuous, any-time decodable point generation. Acknowledging this difference ensures that the proposed framework is correctly categorised as a layer-based scalable codec, directly resolving terminological ambiguities present in general PCC literature.

2.6 Research Gaps and Positioning of This Thesis

Despite significant advancements in PCC, substantial research gaps persist in both traditional and learning-based paradigms, particularly concerning structural fidelity, edge preservation, and density adaptivity. Traditional methods prioritise computational efficiency and simplicity, enabling fast processing for large datasets but at the cost of losing intricate details like local geometries and textures due to rigid hierarchical structures and uniform quantisation [45, 44]. These approaches often introduce artifacts in sparse or irregular regions, failing to adapt to varying point densities and resulting in noticeable quantisation artifacts and lower geometric fidelity for applications requiring high precision, such as cultural heritage digitisation or dense 3D object archival [48, 3].

Early deep learning models have improved upon these by leveraging neural architectures for data-driven feature extraction, achieving better RD trade-offs through end-to-end optimisation [26, 40, 36]. For instance, variational autoencoders and sparse convolutional networks enhance compression ratios while partially addressing geometric redundancies, yet they remain limited in structural fidelity, often over-smoothing fine-grained features and struggling with noise or outliers in sensor-

acquired data [99]. Moreover, these models exhibit deficiencies in edge preservation, where critical boundaries essential for downstream tasks like segmentation are inadequately retained at low bitrates, leading to accuracy degradation measured by metrics such as Hausdorff distance or PSNR [2, 12]. Density adaptivity is another underexplored area, with early methods lacking mechanisms for scalable coding or scalable handling of large-scale clouds (*e.g.*, billions of points from urban scans), resulting in high computational overhead and inefficiency in adaptive streaming for AR/VR or mobile devices [54, 105, 109].

A critical gap lies in the absence of a unified work that holistically addresses these three aspects—structural detail preservation, edge preservation, and density-aware scalable coding—within a cohesive learning-based system. Existing works tend to focus on isolated improvements, such as Transformer-based global attention for structural modelling or edge-centric down-sampling for low-bitrate accuracy, but tend to address them in isolation rather than within a single cohesive framework. This thesis aims to explore complementary strategies across these three aspects, as detailed in the following chapters.

This thesis positions itself to fill these gaps by proposing effective, integrated methods that advance learning-based PCC through structure-preserving, edge-preserving, and density-aware compression strategies. Specifically, this thesis introduce three new approaches detailed in subsequent chapters: Chapter 3 presents a structure-preserving framework that leverages local neighbour aggregation and global attention to maintain fine-grained features and mitigate over-smoothing in irregular data; Chapter 4 develops an edge-preserving technique incorporating cross-domain edge detection and down-sampling for enhanced accuracy under constrained bitrates, ensuring robust performance in downstream tasks; and Chapter 5 proposes a density-aware scalable coding model inspired by scalable video codecs, enabling efficient handling of large-scale clouds with layered transmission for bandwidth-

constrained applications.

Collectively, these contributions present a series of targeted frameworks that address distinct limitations in PCC, demonstrating competitive rate-distortion performance at lower bitrates and improved scalability for bandwidth-constrained applications.

Chapter 3

Transformer-based Geometric Point Cloud Compression with Local Neighbor Aggregation

Extending the literature review in Chapter 2 2, which identified gaps in preserving local-global interactions within point-based and Transformer-driven point cloud compression (PCC) methods, and building on Chapter 1 1's emphasis on RD optimisation for irregular data, this chapter introduces a new Transformer-based approach enhanced by local neighbour aggregation to achieve superior RD performance. This study introduces a geometric Transformer-based approach enhanced by local neighbour aggregation. It begins by introducing the background, defining the problem, outlining the motivations and key contributions of the proposed method, and reviewing relevant literature on Transformer architectures in point cloud processing. The integration of global attention with local feature aggregation is also emphasised here to address limitations in existing techniques. Then, details of the proposed autoencoder-based framework are illustrated, including the encoder with its Global Attention and Local Neighbour Aggregation modules, the decoder for hierarchical reconstruction, and the rate-distortion (RD) loss function. Experimental validation on the ShapeNetCore.v2 [6] dataset is presented, along with comparative evaluations against baseline methods using metrics such as PSNR-D1/D2 and BD-Rate. Finally, the results, including computational complexity and visual comparisons, are discussed and concluded with insights into future directions.

3.1 Introduction

Building upon the foundational concepts introduced in Chapter 1, learning-based autoencoder architectures have become a prominent paradigm in point cloud geometry compression. While traditional standards like MPEG G-PCC [31] provide highly mature and fast baselines via hierarchical partitioning, early learning-based methods seek alternative data-driven representations to further optimise rate-distortion performance at low bitrates. However, these data-driven methods often face challenges in optimally balancing local and global feature interactions, especially during aggressive down-sampling operations like Farthest Point Sampling (FPS) [3, 11].

Learning-based methods offer improvements by learning data-driven features and achieving better RD performance, but they still face limitations in balancing local and global feature interactions, especially in sparse or dynamic datasets [3, 11]. For instance, down-sampling techniques like Farthest Point Sampling (FPS), introduced in PointNet [42], are effective for capturing essential geometry but can lead to local feature loss [138], degrading overall compression quality. While CNN-based approaches excel in local feature extraction, they lack the global context awareness provided by Transformers, and existing Transformer-based methods often prioritise long-range dependencies at the expense of fine-grained local details [71, 36].

With the rapid development of 3D sensors such as LiDAR, the adoption of detailed point cloud models has surged in everyday applications. For instance, static point clouds enriched with attributes such as colour facilitate advancements in virtual reality, augmented reality, mixed reality, and 3D modelling. Meanwhile, large-scale point cloud scenes support critical sectors like robotics and autonomous driving. Despite their promise for realistic 3D representation of objects and environments, point clouds impose significant demands on storage and transmission, particularly for expansive scenes like forests or urban streets, or those requiring

colour information. Moreover, unlike the structured pixels in 2D images, points in these models often exhibit irregularity and high sparsity, complicating processing and analysis. Consequently, downstream tasks such as segmentation, classification, and detection face heightened challenges with large, unstructured datasets. Therefore, developing efficient point cloud coding models is crucial to enhance productivity in both academia and industry [3].

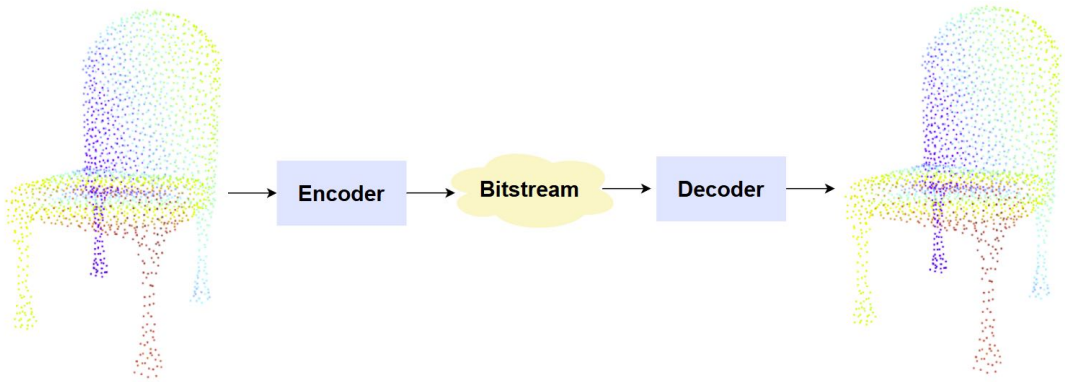


Figure 3.1 : Autoencoder architecture of Learning-based Point Cloud Compression (PCC). A ground truth point cloud converts to a bitstream via an Encoder, and the Decoder decodes the bitstream to get the decoded point cloud model.

Among learning-based PCC methods, autoencoder architectures are widely used as a major backbone to encode and decode point cloud models. As shown in Fig. 3.1, the encoder typically extracts and learns features, followed by an entropy engine to transform these latent features into bitstreams, which are then decompressed by the decoder to regenerate the point clouds. Traditional PCC methods, such as octree-based partitioning (*e.g.*, MPEG G-PCC [31]) and voxel-based approaches [32]), often rely on fixed hierarchical structures that struggle to adapt to irregular point distributions and density variations, leading to inefficiencies, increased bitrates, and artifacts in complex geometries [45, 44].

Learning-based methods offer improvements by learning data-driven features and achieving better RD performance, but they still face limitations in balancing local and global feature interactions, especially in sparse or dynamic datasets [3, 11]. For instance, down-sampling techniques like Farthest Point Sampling (FPS), introduced in PointNet [42], are effective for capturing essential geometry but can lead to local feature loss [138], degrading overall compression quality. While CNN-based approaches excel in local feature extraction, they lack the global context awareness provided by Transformers, and existing Transformer-based methods often prioritise long-range dependencies at the expense of fine-grained local details [71, 36].

This study is inspired by cross-influences from related fields, such as hybrid local-global architectures in image processing and density-preserving strategies in sparse tensor representations [71, 37]. To address these gaps, this study's key contributions include the introduction of a Transformer-based geometric PCC method enhanced by a proposed Local Neighbour Aggregation (LNA) Module. This approach mitigates local-feature losses through a Global Attention Module in the Encoder to extract long-range dependencies from down-sampled point clouds, while the LNA Module—comprising Neighbour Density Embedding and Neighbour Position Embedding layers—captures and aggregates local geometric features, including those from dropped points. Validated on ShapeNetCore.v2 [6], the proposed method achieves at least 30.49% and 23.67% average BD-Rate gains in PSNR-D1 and PSNR-D2 metrics, respectively, with comparable model sizes to other learning-based methods. The following literature review surveys key advancements in learning-based PCC to highlight the specific gaps this study addresses.

3.2 Related Work

Learning-based PCC has evolved rapidly, drawing from advancements in neural networks for 3D data processing. Recent surveys highlight the shift toward data-

driven methods for handling irregularity and sparsity, with significant progress in compression efficiency and quality assessment [73, 3]. These can be broadly categorised into CNN-based and Transformer-based approaches, each addressing different aspects of feature extraction and reconstruction.

3.2.1 CNN-based Methods

CNN-based methods leverage convolutional operations for local feature extraction, often inspired by architectures like Inception Residual Networks (IRN) [89], which use residual mappings for efficient learning. Representative works include Wang *et al.* [60] and Guarda *et al.* [51], which demonstrate improved RD performance through hierarchical feature aggregation. PointNet [42] and PointNet++ [41] further advance this by proposing networks that aggregate local features into global ones, influencing compression techniques like Wang *et al.* [60] and You *et al.* [138]. However, these methods can struggle with global context in highly sparse data, leading to artifacts [3].

3.2.2 Transformer-based Methods

Transformers [139] have gained prominence due to their attention mechanisms, which capture long-range relationships in unstructured points, offering larger receptive fields than CNNs. They can be classified into local and global variants. Global Transformers operate on the entire point cloud to extract dependencies, as in ShapeContextNet [140] (mapping geometry to high-dimensional spaces) and Point-BERT [141] (using stacked attention blocks with max pooling). These excel in tasks like classification and segmentation [142] but may overlook local details post-down-sampling [71]. Local Transformers focus on patch-level aggregation, inspired by PointNet++. The Point Transformer [139] employs hierarchical hierarchical blocks for progressive local feature learning. Recent patch designs address information loss: Density Preserving Point Cloud Compression (D-PCC) [36] uses density patches

with local Transformers for hierarchical aggregationuses density patches with local Transformers for progressive aggregation, while the Patch-based Autoencoder [138] extracts features via PointNet and MLPs. RecentSOTA works like Fu *et al.* [39], Chen *et al.* [115], He *et al.* [36], and Zhang *et al.* [38] show superior RD performance over G-PCC [31] and V-PCC [32], but gaps persist in integrating global attention with local preservation, especially for smaller datasets where efficiency is key [11].

3.3 Methodology

3.3.1 Problem Definition

Following the formal definitions of lossy compression established in Equation 1.1 and Equation 1.2 in Chapter 1, this chapter specifically focuses on optimising the neural mapping $\mathcal{F}_\theta$ to encode the geometric coordinates into a compact bitstream $\mathcal{B}$. The core objective is to design an encoder-decoder pipeline that minimises the rate-distortion cost while explicitly preserving both global structural dependencies and intricate local topological geometries that are traditionally lost during point down-sampling.The problem can be formally defined as follows: point clouds are typically represented as a matrix $\mathbf{X} \in \mathbb{R}^{3\times N}$, where each column $\mathbf{x}_i \in \mathbb{R}^3$ denotes the coordinates of the i-th point, for $i = 1, 2, \ldots, N$. Traditional and learning-based PCC methods optimise an RD trade-off by minimising the loss:

$$\mathcal{L} = \mathcal{D}(\mathbf{X}, \mathbf{X}') + \lambda\mathcal{R}, \tag{3.1}$$

where $\mathcal{D}(\mathbf{X}, \mathbf{X}')$ measures distortion, often via Chamfer distance, between the original $\mathbf{X}$ and reconstruction $\mathbf{X}'$; $\mathcal{R}$ is the bitrate from quantised latents; and λ balances the two. The goal in learning-based PCC is to train a neural model $\mathcal{F}_\theta$ that maps inputs to bitstreams and reconstructs them:

$$\mathcal{F}_\theta : (\mathbf{X}, \mathbf{A}) \mapsto \mathcal{B} \mapsto (\mathbf{X}', \mathbf{A}'), \tag{3.2}$$

with $\mathbf{X}'$, $\mathbf{A}'$ as reconstructed coordinates and attributes, and $\mathcal{B}$ the bitstream.

3.3.2 Overview of the Proposed Method

To address the challenges in point cloud compression, such as preserving local details during down-sampling and capturing global dependencies, this work proposes an autoencoder-based framework that integrates Global Attention and Local Neighbour Aggregation (LNA). This method builds on Transformer architectures to optimise rate-distortion performance, ensuring efficient compression without significant fidelity loss. The overall pipeline, as illustrated in Fig. 3.2, demonstrates how the encoder processes the input through down-sampling and feature aggregation, while the decoder reconstructs via up-sampling and refinement.

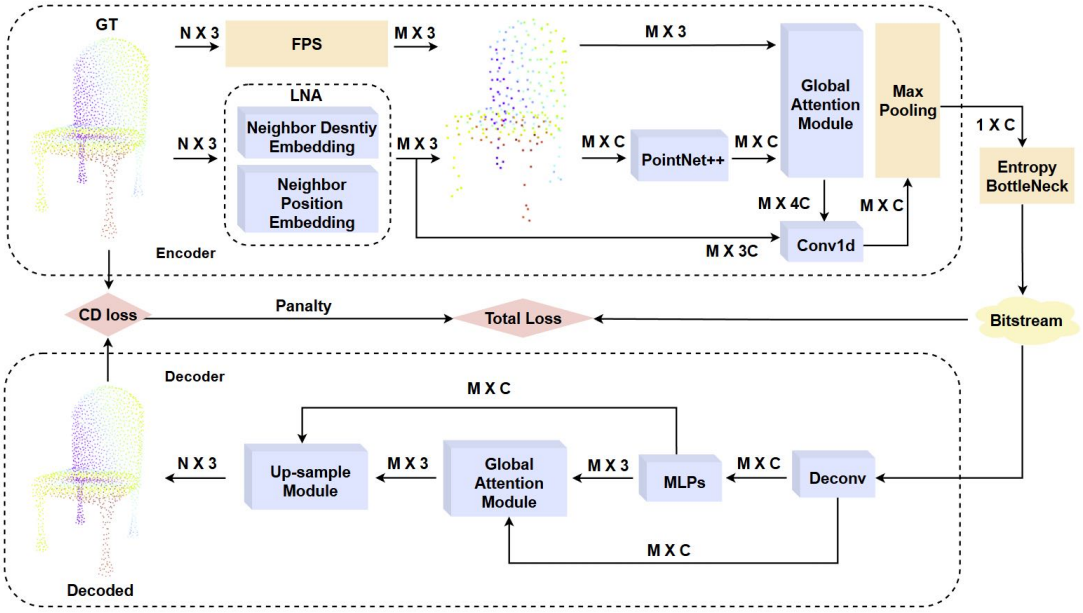


Figure 3.2 : This figure illustrates the pipeline of the proposed method. The workflow begins in the Encoder, where the ground truth (GT) point cloud is down-sampled via FPS, processed through Local Neighbour Aggregation (LNA), and combined with PointNet++ [41] features, fed into Global Attention, then aggregated with Max Pooling and CNNs to prepare the latent for Entropy Bottleneck, yielding the bitstream. In the decoder, the bitstream is up-sampled, passed through Global Attention and MLPs, and refined via Deconv to reconstruct the point cloud. Further technical details are described in Sect. 3.3.2

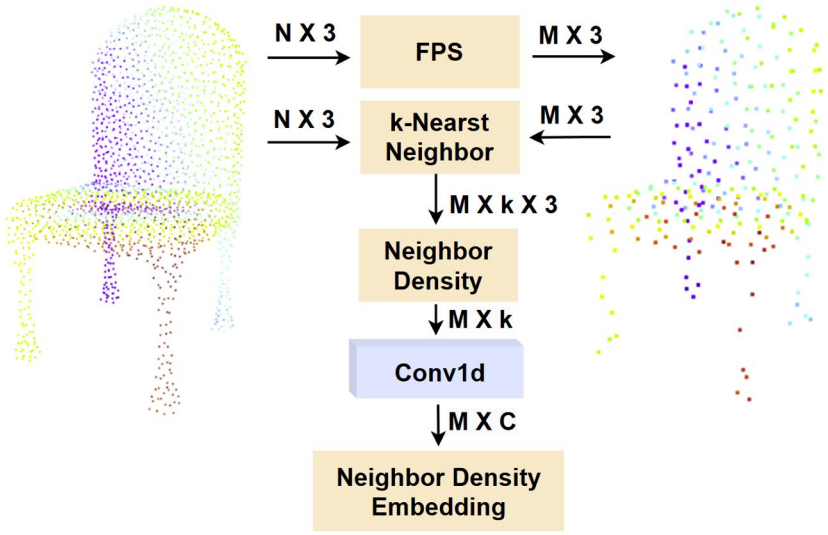
As shown in Fig. 3.2, the proposed method is based on an autoencoder architecture to compress and decompress the point cloud models. Here, N denotes the number of points in the input point cloud, M represents the reduced number of points after down-sampling, and C indicates the feature channel dimension. The Encoder consists of a Global Attention Module and a Local Neighbour Aggregation Module to compress the point clouds into latent features and transform these features into bitstreams by utilising an entropy engine. Then, the decoder learns and decompresses the bitstreams by a deconvolution layer, the same Global Attention

Module, and an up-sample module to decode the point cloud models.

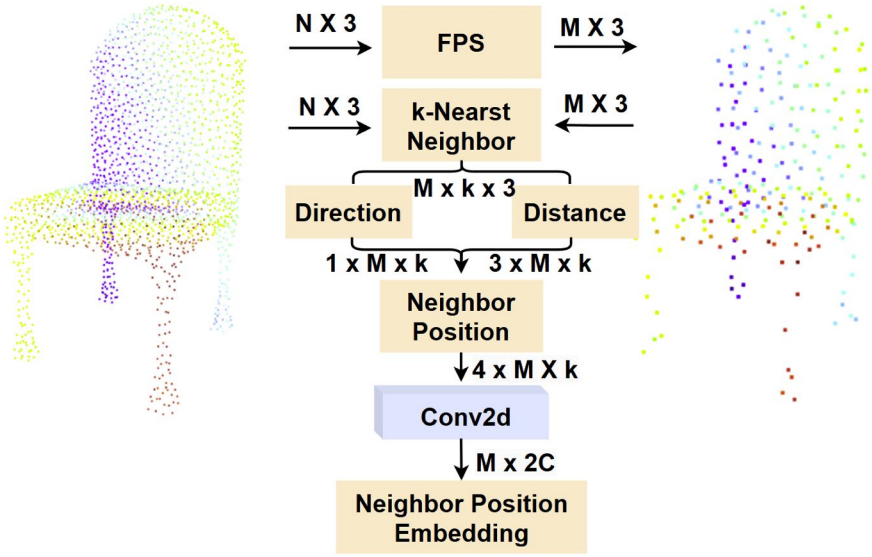
In the Encoder, the original point clouds ($N \times 3$) are down-sampled by FPS to produce a subsampled set ($M \times 3$). Then, the Global Attention Module captures global long-range dependency information from the down-sampled point clouds, while the LNA Module captures local spatial information lost due to down-sampling, including features from dropped points. Their outputs are then concatenated, transformed, and embedded into the entropy engine to obtain the bitstreams. Meanwhile, the decoder uses a deconvolution layer to obtain the decoded down-sampled coordinates and features ($M \times C$), which are augmented by the Global Attention Module and up-sampled by the up-sample module to gradually generate the decoded point cloud models ($N \times 3$). During experiments, $N = 2048$, $M = 128$, and $C = 256$.

3.3.3 Encoder

The Encoder design addresses key challenges in PCC, such as the irregular structure, high data volume, and complex attributes inherent to point clouds, which differ significantly from traditional 2D media [10]. Typically, down-sampling operations like FPS induce geometric information loss and feature leakage at early stages, particularly in high-density regions where down-sampling may overlook critical feature points [36]. To mitigate this and enhance sensitivity to local spatial information, which is often limited in global attention mechanisms, the Local Neighbour Aggregation (LNA) Module is proposed here, as shown in Figures 3.3. This approach draws inspiration from methods that enhance local neighbourhood features, such as densely connecting points within local regions to improve feature aggregation and contextual understanding [143]. By focusing on local aggregation operators, the LNA Module bolsters the model's representational power while maintaining efficiency.



(a) Neighbour Density Embedding Layer



(b) Neighbour Position Embedding Layer

Figure 3.3 : This figure illustrates the architectures of the Local Neighbour Aggregation (LNA) Module. In (a) Neighbour Density Embedding, the process applies FPS to down-sample the input, uses k-NN to find neighbours, computes densities, and embeds via Conv1d to capture local density for compensating down-sampling losses. In (b) Neighbour Position Embedding, it similarly down-samples and finds neighbours, extracts directions and distances, concatenates them, and applies Conv2d to preserve positional and directional details from neighbours. Further technical details are described in Sect. 3.3.3.

This module integrates a Neighbour Density Embedding Layer and a Neighbour Position Embedding Layer to capture local spatial structure, thereby preserving density and geometric details that might otherwise be lost during compression. These layers, detailed in Figures 3.3a and 3.3b, are computed independently and transformed into latent features through convolutional layers, ensuring a robust encoding of local geometry without introducing excessive computational overhead.

a) Neighbour Density Embedding Layer:

Specifically, for Neighbour Density Embedding Layer, consider an original point cloud set $\mathcal{P} = \{p_i\}_{i=1}^{n}$ with n points and a down-sampled set $\mathcal{S} = \{s_j\}_{j=1}^{m}$ with m points. Using a 3D Euclidean distance matrix, for each $s_j \in \mathcal{S}$, k nearest neighbours are selected from $\mathcal{P}$, where each $p_i \in \mathbb{R}^3$, forming a geometry matrix $\mathcal{P}_j \in \mathbb{R}^{m\times k\times 3}$ that captures local details around each down-sampled point. This selection process helps in addressing density variations, as seen in density-preserving approaches that encode local geometry to prevent information loss in sparse or irregular regions [36]. The relative positions are expressed as:

$$\Delta\mathcal{P} = \mathcal{S} - \mathcal{P}_j, \quad \Delta\mathcal{P} \in \mathbb{R}^{m\times k\times 3}, \tag{3.3}$$

where $\mathcal{S}$ is broadcast to match dimensions. Assuming c output channels, the Neighbor Density Embedding E_D is computed as:

$$E_D = \text{conv}(\|\Delta\mathcal{P}\|^2) \in \mathbb{R}^{m\times c}, \tag{3.4}$$

with c denoting the channel dimension. This embedding specifically aims to preserve local density information, which is crucial for maintaining the structural integrity of the point cloud during compression and subsequent tasks, similar to how density-aware networks focus on neighbourhood statistics to enhance feature representation [133].

b) Neighbour Density Embedding Layer:

Considering the relative positions between each down-sampled point and its k neighbours, represented by $\Delta\mathcal{P}$, its Direction $\Delta\mathcal{P}_{\text{Dir}}$ and Distance $\Delta\mathcal{P}_{\text{Dis}}$ are calculated as follows:

$$\begin{cases} \Delta\mathcal{P}_{\text{Dir}} = \|\Delta\mathcal{P}\|_2 \\ \Delta\mathcal{P}_{\text{Dis}} = |\Delta\mathcal{P}| \quad \text{(component-wise absolute values)} \end{cases} \tag{3.5}$$

where $\Delta\mathcal{P}_{\text{Dir}}$ is the Euclidean distance (scalar, reshaped to $1 \times M \times k$), and $\Delta\mathcal{P}_{\text{Dis}}$ captures the absolute differences in each coordinate ($3 \times M \times k$). These are concatenated and processed through a convolutional layer to yield the Neighbour Position Embedding E_P:

$$E_P = \text{Conv}(\text{concat}[\Delta\mathcal{P}_{\text{Dir}}; \Delta\mathcal{P}_{\text{Dis}}]) \in \mathbb{R}^{M \times 2C}. \tag{3.6}$$

This position embedding enhances the model's ability to capture spatial relationships within local neighbourhoods, building on studies of point neighbourhood embeddings that analyse different aggregation methods for neighbouring points to improve robustness and feature learning. The final LNA output, $\mathcal{F}_{\text{LNA}}$, combines these embeddings:

$$\mathcal{F}_{\text{LNA}} = \text{concat}[E_D; E_P] \in \mathbb{R}^{M \times 3C}. \tag{3.7}$$

By integrating both density and position information, the LNA Module provides a comprehensive local feature set that mitigates biases introduced by fixed down-sampling methods and supports better performance in downstream applications.

c) Global Attention Module:

Inspired by the global attention implemented in Guo *et al.* [71] and Zhang *et al.* [38], this work proposes to utilise the Global Attention Module to capture long-range dependencies across point clouds. It consists of four self-attention stages,

where each stage's output feeds into the next. Input features $\mathcal{F}_{\text{in}}$ are embedded into query, key, and value representations Q, $\mathcal{K}$, and $\mathcal{V}$ using learnable weight matrices α, β, and γ:

$$(Q, \mathcal{K}, \mathcal{V}) = (\alpha(\mathcal{F}_{\text{in}}), \beta(\mathcal{F}_{\text{in}}), \gamma(\mathcal{F}_{\text{in}})). \tag{3.8}$$

Then, the self-attention global features $\mathcal{F}_G$ are obtained using normalised attention weights:

$$\mathcal{F}_G = \text{softmax}(Q\mathcal{K}^T)\mathcal{V}. \tag{3.9}$$

To addresses the local feature-capturing limitations of the global attention and local spatial data loss and preserve more local details in Transformers, an offset is applied between the self-attention output features $\mathcal{F}_G$ and the input features $\mathcal{F}_{\text{in}}$, which has been shown to be effective as it sharpens the attention weights and reduces the influence of noise [71]. Here, $\mathcal{F}_{\text{in}}$ comprises down-sampled geometry from $\mathcal{S} \in \mathbb{R}^{M\times 3}$ and enhanced latent features from PointNet++ [41]:

$$\mathcal{F}_{\text{in}} = \mathcal{S} + \text{PointNet}_{++}(\mathcal{S}). \tag{3.10}$$

Specifically, assuming the final output latent features as $\mathcal{F}_{\text{out}}$, the down-sampled point cloud coordinate matrix as $\mathcal{S} \in \mathbb{R}^{M\times 3}$, and φ represents a convolution layer, the final output $\mathcal{F}_{\text{out}}$ can be computed as:

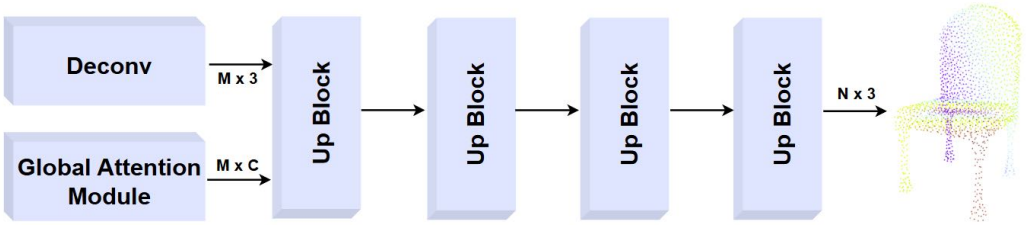
$$\mathcal{F}_{\text{out}} = \mathcal{F}_{\text{in}} + \varphi(\mathcal{F}_{\text{in}} - \mathcal{F}_G). \tag{3.11}$$

This design ensures that the module not only captures semantic affinities across the entire point cloud but also maintains permutation invariance, making it particularly suitable for unordered point data and achieving superior results in tasks like classification and segmentation.

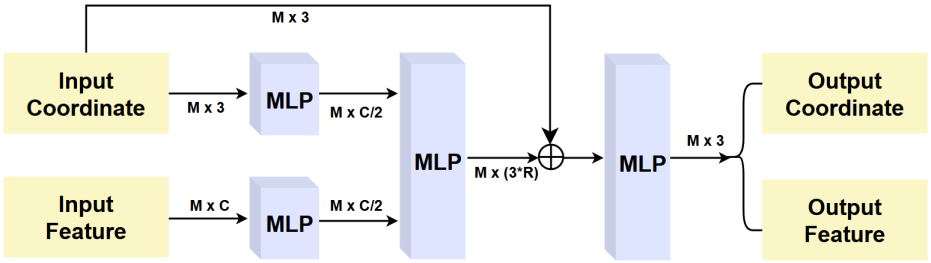
3.3.4 Decoder

In the decoder, the objective is to up-sample and reconstruct the point cloud geometry hierarchicallyreconstruct the point cloud geometry progressively, ensur-

ing that the compressed representation can be efficiently restored while minimising artifacts and preserving structural fidelity. This hierarchical reconstructionprogressive reconstruction is crucial for handling the unordered and sparse nature of point clouds, which can lead to challenges in maintaining density and geometric details during decompression. Following Zhang *et al.* [38], a transposed convolution layer and an MLP layer decode the bitstream into coordinates $\mathcal{S}' \in \mathbb{R}^{m\times 3}$ and features $\mathcal{F}'_{\mathcal{S}} \in \mathbb{R}^{m\times c}$, with $c = 256$, which is shown in Fig. 3.4. The transposed convolution facilitates the expansion of the latent space, effectively reversing the down-sampling process by learning to interpolate missing points based on encoded features, while the MLP refines the feature representations to align with the original data distribution.



(a) Architecture of the Up-sample Layer



(b) Architecture of each Up Block

Figure 3.4 : This figure illustrates the architectures of the Up-sample Layer and each Up Block in the proposed point cloud compression (PCC) method. In (a) Architecture of the Up-sample Layer, the process starts with the Deconv and Global Attention Module applied to the input (corresponding to the decompressed bitstream from Fig. 3.2), followed by a series of Up Blocks for hierarchical up-sampling and refinement, ultimately producing the reconstructed outputpasses through multiple Up Blocks for progressive refinement, applies the Global Attention Module for dependency modelling, and ends with Deconv to produce the output. In (b) Architecture of each Up Block, input features and coordinates are processed via MLPs, combined through multiplication and concatenation, and refined via additional MLPs to generate output coordinates and features. Further technical details are described in Sect. 3.3.4.

This approach is inspired by frameworks that leverage Transformer architectures

for PCC, where the decoder iteratively upsamples pointsthe decoder progressively upsamples points and aggregates features through attention mechanisms to achieve accurate reconstruction. The same Global Attention Module refines and improves an Up-sample Module, illustrated in Fig. 3.4a, reconstructs the point cloud $\mathcal{P}' \in \mathbb{R}^{n\times 3}$ by 4 up-sample blocks sequentially. These four identical Up-sample Blocks, each with four MLP layers balancing input coordinates and features, with an Up-sample rate $R = 2$, as shown in Fig. 3.4b. Outputs cascade through subsequent blocks, mirroring the Global Attention Module's staged structure, which promotes hierarchical feature refinement similar to density-preserving methods that adaptively up-sample based on latent features to maintain original geometry and prevent distortions in sparse areas. This design not only improves RD performance but also supports scalability for large-scale point clouds by incorporating joint sampling and feature extraction in end-to-end frameworks.

3.3.5 Loss Functions

A standard RD Loss, which is commonly adopted in baseline SOTA learning-based PCC methods such as D-PCC [36], IPDAE [138], and various autoencoder-based approaches [3, 60], is employed here to preserve original geometry details. The above works utilise similar formulations to balance compression efficiency and reconstruction quality, enabling end-to-end optimisation for variable-rate scenarios. This study's choice aligns with this paradigm to facilitate fair comparisons and leverage proven effectiveness in minimising artifacts while controlling bitrates:

$$L = \lambda D + R, \tag{3.12}$$

where λ balances distortion D and bitrate R. The bitrate R is derived from the entropy engine, while D uses the Chamfer Distance [119] between the reconstructed

point cloud $\mathcal{P}' \in \mathbb{R}^{N\times 3}$ and ground truth $\mathcal{P} \in \mathbb{R}^{N\times 3}$:

$$D_{(\mathcal{P}',\mathcal{P})} = \frac{1}{N}\left(\sum_{x\in\mathcal{P}'}\min_{y\in\mathcal{P}}\|x-y\|_2^2 + \sum_{y\in\mathcal{P}}\min_{x\in\mathcal{P}'}\|y-x\|_2^2\right), \tag{3.13}$$

where N is the number of points, and x and y denote points in $\mathcal{P}'$ and $\mathcal{P}$, respectively. Chamfer Distance is selected as the distortion metric, because it symmetrically measures average nearest-neighbour distances, providing a robust quantification of shape dissimilarity that is particularly effective for geometry compression and reconstruction tasks [36]. This loss formulation supports variable-rate compression by adjusting λ, enabling better overall performance tailored to a hybrid global-local architecture.

3.4 Experiments

3.4.1 Dataset

The proposed method is experimentally validated on the ShapeNetCore.v2 dataset [6], a widely recognised benchmark in the field of 3D point cloud processing and compression, as evidenced by its adoption in prominent SOTA works including D-PCC [36], IPDAE [138], and other learning-based frameworks [60, 3]. This dataset is specifically selected as it offers a massive collection of dense, object-centric models (over 51,127 3D point clouds spanning 55 categories, such as cars, tables, laptops, and chairs) that enable robust evaluation across varied shapes and complexities while maintaining computational tractability. ShapeNetCore.v2 is selected to leverage to ensure direct comparability with prior methods that report results on similar subsets, allowing us to demonstrate superior BD-Rate gains in a standardised setting. Each point cloud model was uniformly sampled to 2,048 points using FPS, a standard preprocessing step in baselineSOTA papers [36, 138, 41], which iteratively selects distant points to preserve overall topology and even distribution, reducing redundancy and facilitating efficient training. This comprehensive, category-diverse

testbed highlights the advantages of the proposed local-global hybrid approach for high-fidelity 3D object compression. This setup justifies this study's choice by providing a comprehensive, category-diverse testbed that highlights the advantages of the proposed local-global hybrid approach over methods evaluated on denser or sparser datasets.

3.4.2 Implementation and Training Details

The proposed model was trained and tested on a computer with an i7-12700K CPU at 3.61GHz, 32 GB of main memory, and an NVIDIA GeForce RTX 3080 GPU. During training, we set λ ranging from 1 to 1000 to control the RD trade-off, where smaller values prioritise lower bitrates and larger ones emphasise reconstruction quality, allowing the model to adapt to various compression scenarios. We also set the initial learning rate to 0.001, batch size to 32, and used Adam optimisation. Following the training strategy of Guarda *et al.* [1], we set the patience to 5 epochs for the early stopping of the training process to prevent overfitting if the validation loss does not decrease for up to 5 epochs, and set the CD loss penalty to range from 20 to 1000.

3.5 Results and Discussion

To effectively validate the proposed method, a comparative study is conducted here with baselineSOTA learning-based point cloud geometry compression techniques, and the result is shown in Fig. 3.5.

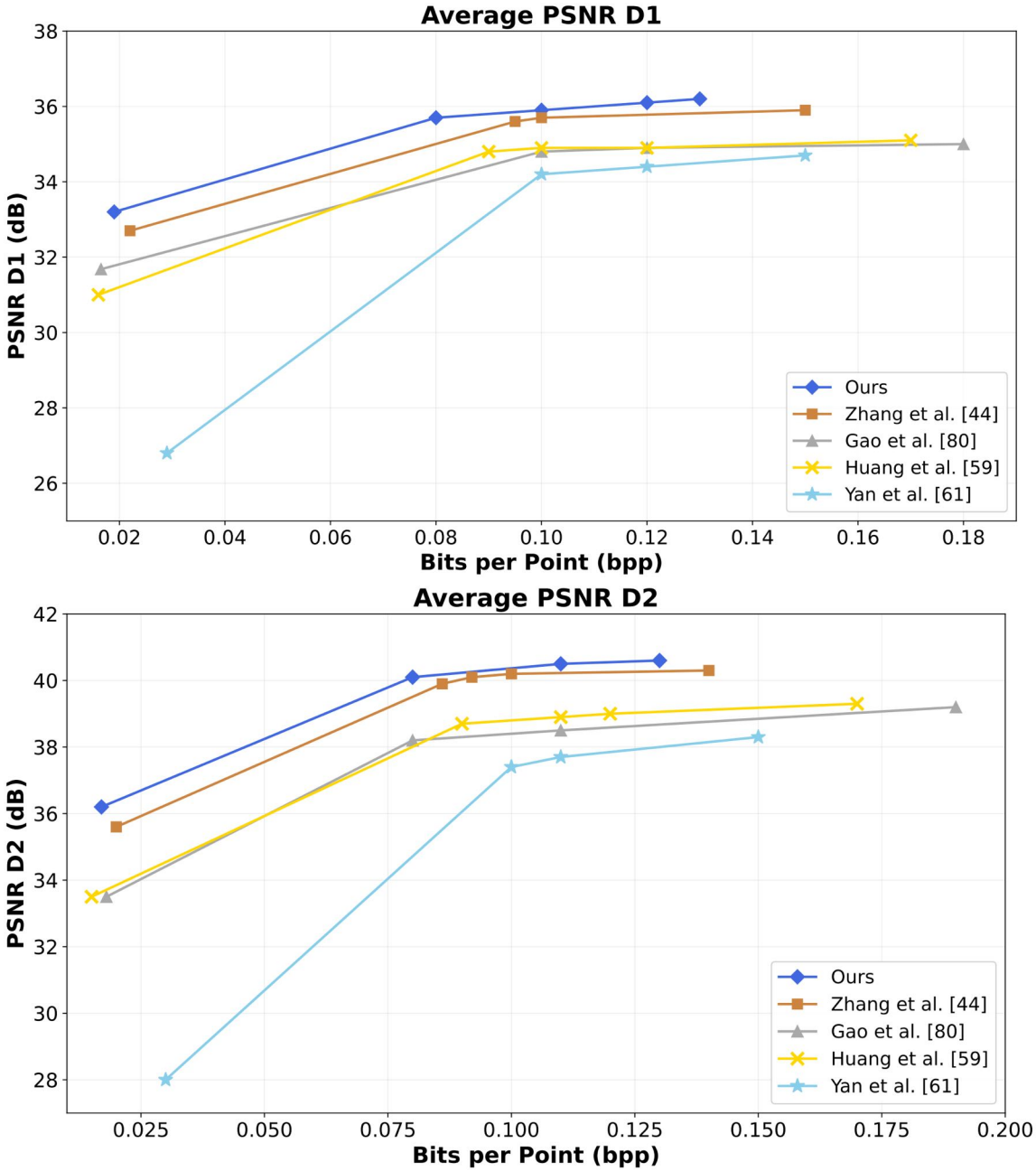


Figure 3.5 : Objective comparison result of the average PSNR-D1 and average PSNR-D2 in ShapeNetCore.v2 [6].

In details, the results include the methods of Zhang *et al.* [38], Yan *et al.* [56], Huang *et al.* [55], and Gao *et al.* [77]. PSNR-D1 & D2 and BD-Rates are selected to compare the results quantitatively.

3.5.1 Quantitative Performance

To assess the efficacy of the proposed Transformer-based PCC method, this study employs established metrics from the JPEG Pleno point cloud coding common test conditions [49], which are widely adopted in the literature for objective quality evaluation in geometry compression tasks [3, 45]. These metrics enable standardised comparisons with baseline methods, focusing on both reconstruction fidelity and compression efficiency. Specifically, this study evaluates using Peak Signal-to-Noise Ratio (PSNR) computed over point-to-point (D1) and point-to-plane (D2) distances, as well as Bjøntegaard Delta Rate (BD-Rate), which quantifies bitrate savings at equivalent quality levels. These choices are justified by their prevalence in baselineSOTA works such as D-PCC [36], IPDAE [138], and Transformer-based approaches [38, 39], allowing direct benchmarking while highlighting the proposed method's advancements in preserving geometric details with reduced bitrates.

1) PSNR-D1 and PSNR-D2:

Following JPEG Pleno guidelines [49], this study computes PSNR as the ratio of the peak signal (bounding box diagonal) to the root-mean-square error derived from D1 and D2, respectively. This dual-metric approach provides a comprehensive view of fidelity, addressing limitations of single-metric evaluations that may overlook either local or structural errors [144].

As illustrated in Fig. 3.5, which plots average PSNR-D1 and PSNR-D2 against bits-per-point (BPP) across the ShapeNetCore.v2 [6] test set (10,261 models), this study's method consistently outperforms baselineSOTA baselines including Zhang *et al.* [56], Gao *et al.* [77], Huang *et al.* [55], and Yan [56]. At low bitrates (*e.g.*, 0.03–0.09 BPP), the proposed curve exhibits steeper gains, achieving up to 2–4 dB higher PSNR-D1 and PSNR-D2 compared to competitors, indicating superior low-

rate performance where bandwidth is constrained. For instance, at 0.12 BPP, the proposed method reaches approximately 364 dB in PSNR-D1, surpassing Zhang *et al.* by 2 dB and Gao *et al.* by 1.5 dB. Similarly, in PSNR-D2, this study attains 38 dB at the same rate, outperforming Huang *et al.* [55] by 3 dB. This superiority stems from the hybrid global-local architecture, which mitigates down-sampling losses via LNA and captures long-range dependencies through offset attention, resulting in fewer artifacts and better preservation of intricate geometries. Overall, across the 51,127 models in ShapeNetCore.v2, this proposed method demonstrates higher average PSNR values at equivalent bitrates, requiring 20–40% less BPP to match baselineSOTA quality, underscoring its efficiency for diverse object categories.

2) BD-Rate:

To further quantify compression efficiency, this study utilises BD-Rate, a standard metric derived from the Bjøntegaard model [121], which computes the average bitrate savings (in BPP) relative to a reference method while maintaining equivalent PSNR levels. Negative BD-Rate values indicate bitrate reductions, with larger magnitudes signifying greater efficiency. This metric is particularly convincing as it integrates RD curves over a range of operating points, providing a holistic comparison beyond isolated BPP-PSNR pairs [45]. This study benchmarks against prominent learning-based SOTA methods, selected for their relevance to Transformer and autoencoder paradigms, ensuring apples-to-apples evaluation on the same dataset and conditions.

Table 3.1 : BD-Rate Evaluation Compared with BaselineSOTA Methods

Method Comparison	**BD-Rate (PSNR-D1)**	**BD-Rate (PSNR-D2)**
Ours vs Zhang *et al.* [38]'s	−30.49%	−23.67%
Ours vs Yan *et al.* [56]'s	−60.99%	−60.40%
Ours vs Huang *et al.* [55]'s	−76.41%	−77.84%
Ours vs Gao *et al.* [77]'s	−41.96%	−48.70%

As detailed in Table 3.1, our method achieves substantial BD-Rate reductions, saving at least 30.49% in bitrate for PSNR-D1 and 23.67% for PSNR-D2 compared to the closest Transformer-based competitor (Zhang *et al.* [38]). Against earlier CNN-centric methods like Yan *et al.* [56] and Huang *et al.* [55], gains exceed 60–77%, reflecting the proposed architecture's enhanced feature learning via global attention and local aggregation, which better handles sparsity and irregularities than voxelised or pure CNN approaches. Compared to Gao *et al.* [77], a local-focused method, hybrid design yields 41.96–48.70% savings, validating the integration of LNA to address local losses that pure global or local models overlook. These results are averaged over the diverse ShapeNetCore.v2 categories, demonstrating robustness across geometries (*e.g.*, higher gains on complex shapes like airplanes vs. simpler ones like tables). In summary, these metrics convincingly establish the proposed method's superiority in generating high-fidelity reconstructions at lower bitrates.

3) Compared with Traditional Geometry Methods:

Table 3.2 : Comparison with G-PCC

Method Comparison	**Rate (BPP)**	**PSNR-D1 / PSNR-D2**
G-PCC [6]	0.72	39.13 / 46.60
Ours	0.064	42.18 / 46.60

Because ShapeNetCorev2 contains normalised point clouds, this study follows the testing strategy from Zhang *et al.* [38] and compares this work with G-PCC for the same test models, which achieve close PSNR values in G-PCC for detailed comparison. As detailed in Table 3.2, our method achieves equivalent or superior reconstruction quality—matching PSNR-D2 at 46.60 dB while surpassing PSNR-D1 by 3.05 dB (42.18 vs. 39.13)—at a dramatically lower bitrate of 0.064 BPP compared to G-PCC's 0.72 BPP.

4) Computational Complexity:

Table 3.3 : Computational Complexity Compared with Baseline MethodsSOTA

Method Comparison	**Model Size**	**Decoding Time for Each Model (s)**
Ours	44.508 MB	1.049
Zhang *et al.* [38]	42.991 MB	0.424

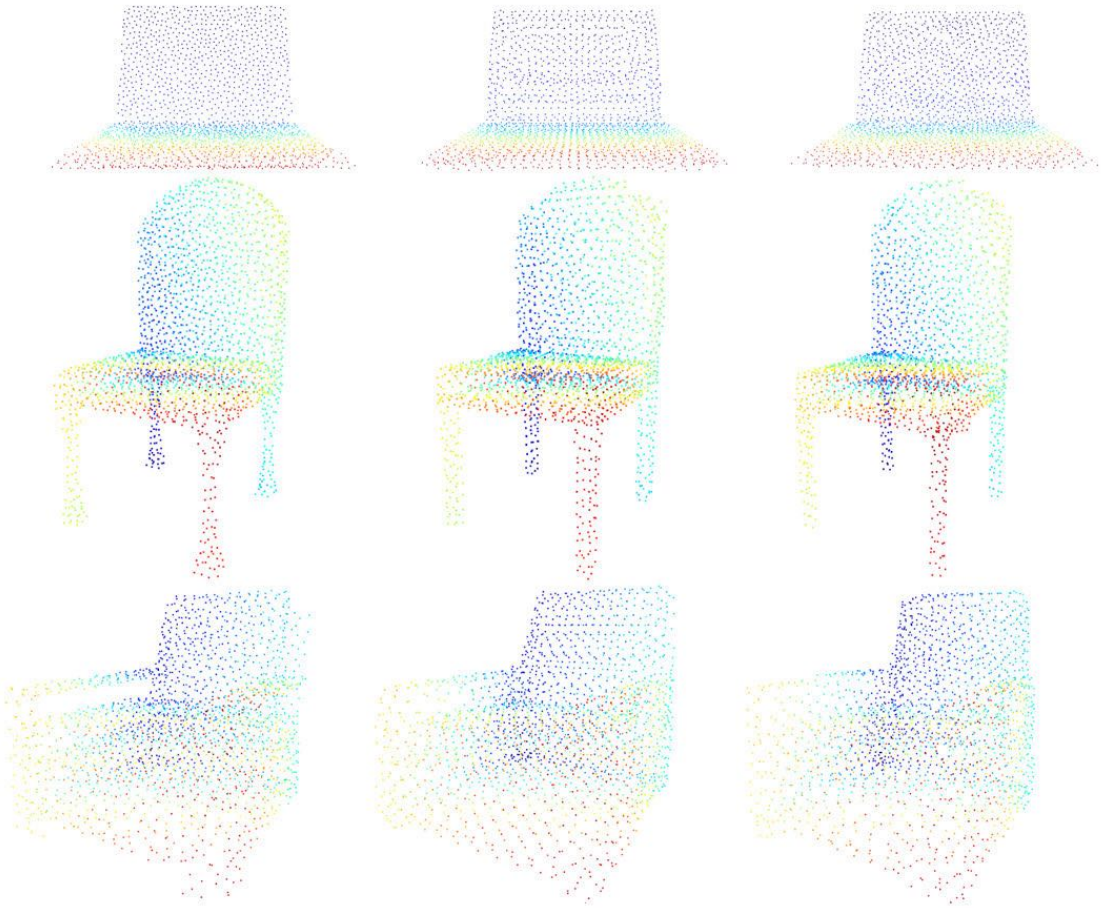
As detailed in Table 3.3, the model size of the proposed method (44.508 MB) is comparable to that of the baseline by Zhang *et al.* [38] (42.991 MB), exhibiting only a marginal 3.5% increase in parameters. Regarding computational latency, the proposed method requires an average decoding time of 1.049 seconds per model, compared to 0.424 seconds for the baseline. This increased decoding time is an anticipated architectural trade-off; the integration of the Local Neighbour Aggre-

gation (LNA) module introduces necessary computational overhead during feature extraction to rigorously preserve high-fidelity local geometries. However, as demonstrated in Table 3.1, this additional computational cost yields a substantial 30.49% BD-Rate improvement in PSNR-D1. For target applications such as high-fidelity 3D asset archival, where structural integrity is prioritised over strict low-latency decoding, this trade-off is highly advantageous.

Evaluating the feasibility of a comparison based on equal latency reveals significant challenges within the current framework. Constraining both methods to identical decoding times is not straightforward because the LNA module operates as an integral architectural component. Specifically, its feature aggregation mechanism is tightly coupled with the reconstruction pipeline of the decoder. This module cannot be partially disabled or throttled during inference without fundamentally altering the learned representations, an action that would necessitate comprehensive retraining under a distinct architecture. Consequently, conducting a controlled evaluation under equal time constraints would require engineering a lightweight LNA variant characterized by reduced neighbor counts or fewer aggregation layers. Such modifications constitute a separate architectural exploration falling outside the scope of this current study. Exploring these variants constrained by latency remains a highly promising direction for future research, particularly regarding early exit decoding strategies that exchange marginal reconstruction quality for accelerated processing speeds. From the results shown in Table 3.3, the size of the model of this work is slightly larger than the methods of Zhang *et al.* [38], while this study can decode the point cloud models quicker among the randomly selected 10260 testing point clouds. Specifically, the proposed method is 3.5% bigger than Zhang *et al.* [38], and decoding time for each model is reduced by around 60%.

3.5.2 Visual Comparison

To better illustrate the results, visual comparisons are provided in Fig. 3.7. While macroscopic differences at extreme low bitrates appear subtle due to the high overall compression quality, closer inspection reveals that the proposed approach tends to preserve marginally more regular and continuous structures in highly curved local regions. Furthermore, along the boundary edges, the reconstructions exhibit slightly fewer outlier artifacts compared to the baseline method by Zhang *et al.* [38], visually corroborating the quantitative PSNR improvements.



(a) (b) (c)

Figure 3.6 : Visual comparison of reconstructed point cloud models. Columns from left to right: (a) the baseline approach by Zhang *et al.* [38], (b) the proposed approach, and (c) the Ground Truth. The proposed method demonstrates subtle improvements in maintaining local structural boundaries.

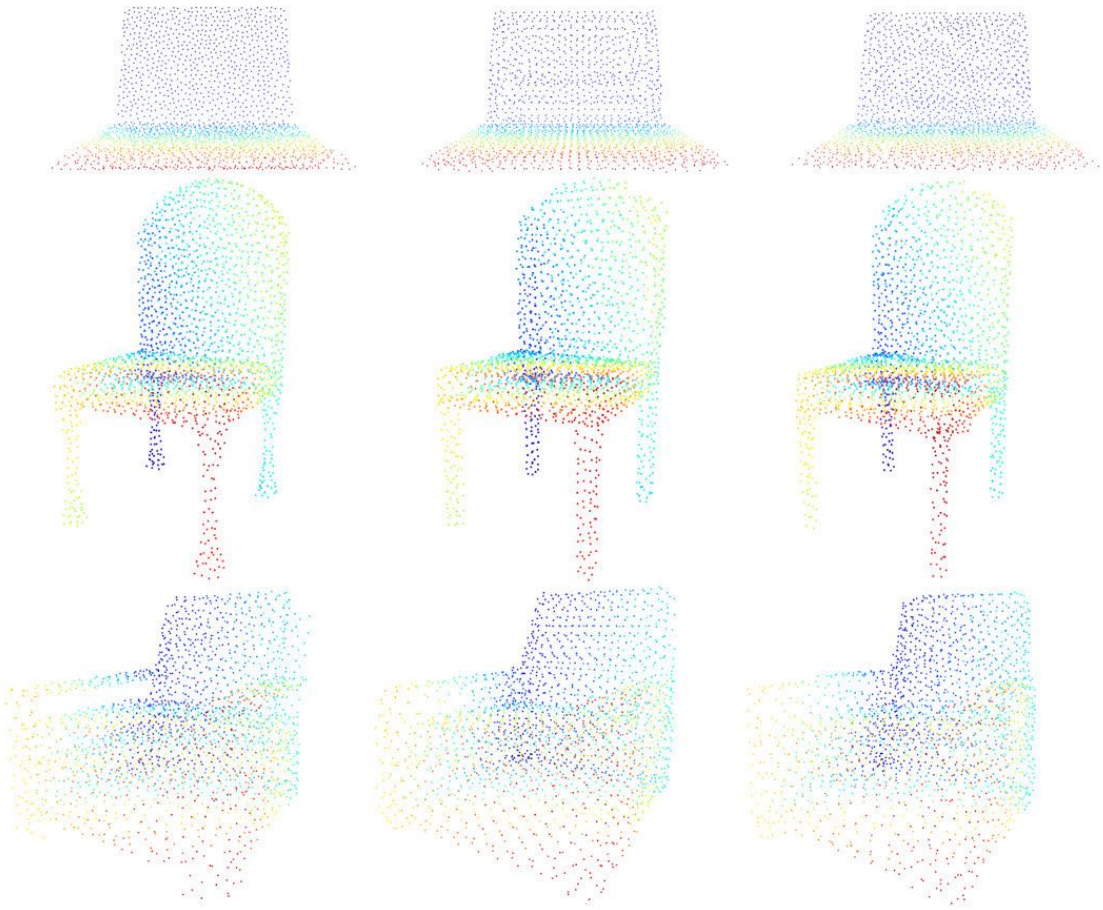


Figure 3.7 : Visual comparison of the point cloud models decoded by our approach (the middle column) and those decoded by the approach in Zhang *et al.* [38]. As can be seen, the result of this work includes more local boundary details of point clouds.

3.6 Summary

This section proposes a Transformer-based PCC method with a Local Neighbour Aggregation Module. This study utilises a proposednovel Local Neighbour Aggregation Module to capture local spatial information and uses a Global Attention Module to learn the down-sampled global features. The superior performance of the proposed method has been validated by comparing it with prominentSOTA learning-based methods on the benchmark dataset ShapeNetCore.v2. For the same

bit rate, this work is better at preserving point cloud details with higher PSNR-D1 and PSNR-D2. In the future, the optimal integration or trade-off between Global Attention and Local Feature Aggregation might be a promising topic to be examined. Additionally, reducing the decoding latency introduced by the LNA module by lightweight neighbour aggregation variants, pruned attention heads, or early-exit decoding strategies can represents an important avenue for improving the method's applicability in latency-sensitive scenarios. Furthermore, evaluating the proposed method on additional datasets, including sparse LiDAR benchmarks such as SemanticKITTI, would further validate its generalisability across diverse point cloud characteristics.

Chapter 4

Compressed Point Cloud Classification with Point-based Edge Sampling

Leveraging the Transformer-based compression framework from Chapter 3, which improves geometric fidelity at low bitrates, and addressing the downstream task limitations identified in Chapter 2's review (*e.g.*, accuracy degradation in compressed domains), this chapter introduces compressed point cloud classification (CPCC) with attention-based edge sampling to preserve structural details. Here, this chapter presents a proposed framework for CPCC that incorporates attention-based edge sampling to preserve critical structural information, enhancing accuracy at reduced bitrates. This chapter first introduces the background, highlights motivations and contributions, defines the problem, and reviews literature on down-sampling techniques and compressed-domain analysis. Then it articulates how attention-based edge sampling addresses gaps in existing methods, such as the oversight of salient outlines in traditional sampling approaches and limited adaptability in voxel-based frameworks. The proposed CPCC-PES architecture is detailed, including the encoder with neighbour attention and edge sampling modules, an entropy bottleneck with gain layers, and a lightweight decoder. Experimental validation on benchmark datasets demonstrates superior BD-Rate savings and Top-1 accuracy. This chapter concludes with discussions on results, ablation studies, visualisations, and future directions, offering insights into scalable compressed point cloud analysis for edge devices.

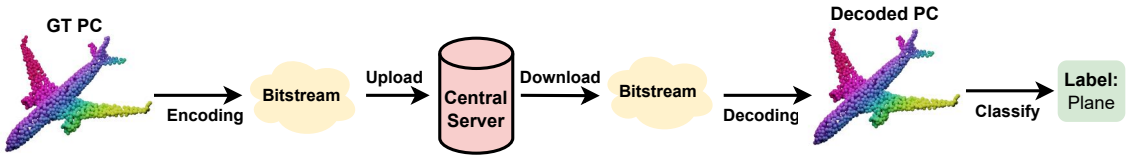
4.1 Introduction

Building upon the geometric compression foundations established in Chapter 3, this chapter addresses the specific challenges of downstream analysis in bandwidth-constrained environments. As highlighted in Chapter 2, performing high-level vision tasks on uncompressed or fully decompressed 3D data often requires significant transmission bandwidth and computational overhead [3].

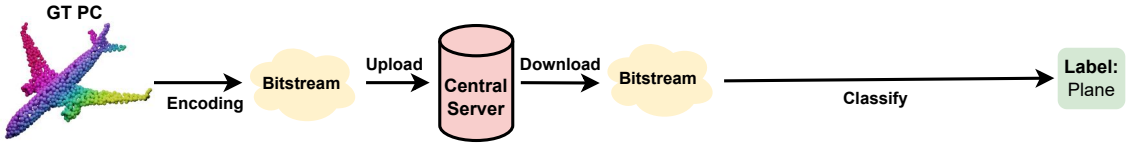
Typically, a conventional strategy for handling this is Decompressed Point Cloud Analysis (DPCA). In DPCA, point clouds are analysed and compressed into a bitstream on the scanning or server-side terminal. The bitstream is then transmitted to the central computing hub to be fully decompressed into the decoded point cloud. Then, the computing hub conducts downstream vision tasks and dispatches the analysed results back to the end devices. While this strategy preserves the overall geometry, it requires high computational resources and stable transmission mediums. Furthermore, decoded point clouds often exhibit artifacts such as outliers or shrinkage, which can degrade the performance of downstream tasks compared to using uncompressed originals [11].

To solve the above issues, this chapter focuses on Compressed Point Cloud Analysis (CPCA), which identifies a more feasible, economically-friendly, and effective bitrate-saving solution to process the downstream tasks. As shown in Fig. 4.1a, instead of utilising the decoded point cloud information to do point cloud classification in a manner of Decompressed Point Cloud Classification (DPCC), CPCC conducts the point cloud classification directly on the encoded bitrate. Existing approaches, such as [2] and [1], have revealed the potential and effectiveness of CPCC. However, their classification performance still falls behind the normal baseline point cloud classification results. Powerful PCC encoders and enhancement algorithms can be adapted and significantly improve the CPCC result.

Another gap hindering the effectiveness of CPCA is that point clouds are big and irregular. In traditional 3D point cloud analysis, widely-used mathematical sampling methods include Farthest Point Sampling (FPS) [61], Random Point Sampling [145], and Grid Sampling [146]. Embedded with deep learning techniques such as Convolution Neural Networks and Transformers, these down-sampling methods can play a significant role in capturing and providing the essential structural information of a point cloud for improving the performance of learning-based downstream tasks efficiently. To add more adaptability to different downstream tasks, some advanced learning-based down-sampling methods have been proposed, tailored to different downstream tasks, *e.g.*, Sample-Net [57], Skeleton-aware Down-sampling [147], DA-Net [148], and LightN [58]. Even though many advanced learning-based techniques have been applied in utilising the representative semantic information of point clouds, many of these have not considered the shape outlines and edge information as special and representative features of the point cloud.



(a) The general architecture of Decompressed Point Cloud Classification (DPCC).



(b) The general architecture of Compressed Point Cloud Classification (CPCC).

Figure 4.1 : The general architectures of DPCC and CPCC.

The proposed method leverages the cross-influence from attention in point cloud analysis and proposes a proposed CPCC approach with a point-based edge sampling module that focuses on preserving the edge information of the point cloud for better

classification performance while maintaining a smaller bitrate. Edge information is widely used in 2D image classification and segmentation because it can capture crucial representative semantic shape contour information [149]. Whereas, in the field of the point cloud processing, even though many advanced techniques have been applied in storing representative semantic information for the point cloud in a learnable way, many of them have not considered the shape outlines and edge information as special and representative features of the point cloud [150]. This work extends the success of 2D edge-preserving processing into the 3D point cloud domain and performs edge-sampling. Because it is difficult to detect the edges of point clouds efficiently through a sampling method, since point clouds are often irregular and contain regions of different density and sparsity, a learning-based sampling method is utilised here to solve the problem.

The result shows that our model has a superior rate-accuracy trade-off performance than the baseline work. Besides, following the prior work, a new evaluation metric is proposed here to evaluate the trade-off performance between bitrate and Top-1-Accuracy in CPCC, named BD-T1-Accuracy, for future research exploration. Specifically, this work can achieve over 90% Top-1 Accuracy with only 0.08 bits-per-point (BPP) consumption, which consumes only 20% bitrate compared with the CPCC baseine work. Besides, it has over 96% reductions in BD-Rate over specialised DPCC codecs on the ModelNet40 [6] dataset.

4.2 Literature Review

4.2.1 Compressed Point Cloud Classification

Building on the role of down-sampling in point cloud processing—as discussed in the previous subsection, where learning-based methods enhance task-specific feature retention—the integration of such techniques into compression frameworks is crucial for CPCA. This section reviews the evolution from general point cloud analysis

(PCA) to specialised CPCA and Decompressed point cloud analysis (DPCA), highlighting how efficient down-sampling contributes to maintaining semantic integrity in compressed domains, which is central to the chapter's focus on CPCC.

Since the introduction of PointNet [42] and PointNet++ [41], PCA based on point-wise information has become foundational in processing irregular 3D data without preprocessing like voxelisation. This end-to-end architecture has inspired numerous supervised and self-supervised methods that employ advanced operations, such as point convolutions (*e.g.*, PointConv [151], KPConv [76], and PointConvFormer [152]) and attention mechanisms (*e.g.*, Point Transformer [139], Point Cloud Transformer [71], Geo-former [153], Point4Transformer [154], and Point Transformer v2 [146]), as well as graph-based approaches like DGCNN [67] with dynamic EdgeConv blocks.

The growing demand for efficient end-device processing has led to the emergence of CPCA, which operates directly in the compressed domain to save resources for tasks like classification. For instance, Deep Learning-Based Compressed Domain Point Cloud Classification [155] uses a bridge layer to link the latent representations from JPEG Pleno Point Cloud coding Verification Model [1] to a Point Grid Partial Classifier [156] in a voxel-based manner. Similarly, Learned Point Cloud Compressed Classification (LPCCC) [2] develops an end-to-end network based on PointNet [42], incorporating two gain layers in the entropy bottleneck to enhance adaptability and stability for latent geometry coding while achieving high accuracy. In contrast, DPCA methods, such as that by Bojun *et al.* [157], employ a gradient bridger with point-matching to propagate gradients from decoded point clouds to detection networks.

Despite their potential in outperforming traditional methods with bitrate savings, existing CPCA and DPCA approaches remain limited. Current research pri-

marily addresses classification and detection, leaving opportunities for exploration in other applications like segmentation or registration. Moreover, performance can be enhanced by integrating advanced PCA modules, such as attention mechanisms [70], into frameworks like those in LPCCC [2]. Inspired by the attention-driven refinements in learning-based down-sampling, which will be discussed in Sec. 4.2.2, and PCA's focus on semantic relationships, this work addresses the key challenge of edge information loss in low-bitrate compression, introducing an attention-based edge sampling to preserve structural details and boost classification accuracy in CPCC.

4.2.2 Point Cloud Down-sampling

Down-sampling is a fundamental step in PCA because it reduces the number of points while preserving the overall distribution or structure of the original point cloud using a small point subset, thereby enabling efficient processing for point cloud downstream tasks such as compression.

Generally, traditional down-sampling approaches are mainstream because they can be easily integrated with deep learning techniques by offering a consistent input size to the next encoder stage. For example, Farthest Point Sampling (FPS) [61] has been widely used in many popular PointNet [42] variant methods to preserve the general representative information and structure of the point cloud. These variant methods include PointNet++ [41], Point-MAE [158], PT [139], PCT [71], and Stratified Transformer-3D [159]. As another example, Grid-sampling is used in Point Transformer v2 [146] to support point neighbour pooling when performing point cloud segmentation. In detail, grid-sampling performs the initial down-sampling to create spatially aligned partitions, while point neighbour pooling aggregates features (*e.g.*, via max-pooling) from neighbouring points within those partitions or groups, enabling more efficient and aligned feature extraction compared to traditional ap-

proaches.

While traditional methods, such as FPS or voxel grid filtering, offer a consistent and straightforward approach to reducing point counts in point clouds, they often lack adaptability to task-specific characteristics, relying instead on fixed, task-agnostic geometric rules. For instance, as shown in SampleNet [57], techniques like random or FPS sampling can discard semantically important points, resulting in degraded downstream performance for tasks such as classification or reconstruction. Similarly, REPS [160] emphasises that these rigid, non-learnable strategies limit optimisation for specific goals, such as maintaining reconstruction fidelity or classification accuracy. This is further evidenced in Almost-Universal yet Task-Oriented Sampling [161], where traditional samplers are outperformed by learned alternatives due to their failure to incorporate task-aware priors, often leading to redundant or uninformative point selections across applications.

To overcome these limitations, learning-based down-sampling methods leverage neural networks to select points that are most representative for a given task, enabling adaptive and optimised sampling. A foundational example is S-Net [162], which uses a global representation of the point cloud to generate new geometry coordinates, effectively capturing underlying structures for tasks like classification (*e.g.*, on ModelNet40 [6]) and reconstruction (*e.g.*, via Chamfer distance metrics). Building directly on S-Net's idea of task-driven point generation, SampleNet [57] introduces a differentiable module that learns to predict subsets of points by optimising downstream performance, such as in classification or reconstruction scenarios. This progression inspired further refinements: PST-NET [163] enhances the approach by integrating attention mechanisms to prioritise relevant features during sampling, while LightN [58] draws from these to create a lightweight Transformer framework that balances computational efficiency with high performance. Additionally, some methods emphasise preserving semantic information; for example, Skeleton-aware

Down-sampling [147] exploits the medial axis to guide sampling, ensuring retention of key structural elements like edges and boundaries, which are vital for accurate classification and segmentation.

When comparing traditional and learning-based down-sampling methods, several key dimensions highlight their differences, supported by empirical evidence from recent studies. In terms of computational efficiency, traditional methods like FPS or voxel grid sampling are generally faster and lighter, with FPS achieving $O(nlogn)$ time complexity for uniform distribution preservation, making them suitable for real world applications. However, learning-based approaches, while potentially more computationally intensive during inference due to neural network evaluations, offer long-term efficiency gains through better downstream task performance, as demonstrated in a comparative study on plant phenotyping where learned methods reduced overall processing time by optimising feature retention [164]. Regarding flexibility, traditional methods are rigid and task-agnostic, often failing to adapt to varying data distributions or specific objectives, leading to loss of critical features like sharp edges [165]. In contrast, learning-based methods provide high flexibility by incorporating task-specific priors, resulting in superior adaptability and robustness to noise or outliers, with SampleNet showing up to 10-20% improvements in classification accuracy over FPS on benchmarks like ModelNet40 [57]. Finally, on feature preservation, traditional approaches prioritize global integrity but struggle with semantic details, whereas learned methods excel in retaining task-relevant structures, though they may introduce artifacts like holes in sparse regions if not carefully designed [25]. Overall, while traditional methods excel in simplicity and speed for general use, learning-based ones dominate in complex, task-oriented scenarios, as evidenced by their consistent outperformance in deep learning pipelines for point cloud tasks.

Motivated by these advantages, this work adopts a learning-based down-sampling

strategy to address the challenges in compressed point cloud classification (CPCC), where preserving edge information is crucial for maintaining geometric fidelity at low bitrates. Traditional methods often discard boundary points critical for classification accuracy in compressed scenarios, as seen in limitations highlighted by REPS and SampleNet. This work is inspired by the attention mechanisms in PST-NET and LightN, which enable adaptive feature prioritisation, to develop an attention-based edge sampling module. This approach computes normalised correlation maps via attention to detect high-variability points (edges) using standard deviation, ensuring task-specific retention of structural details for efficient compression and superior classification performance in CPCC, as validated on datasets like ModelNet40 [6].

4.3 Methodology

4.3.1 Problem Definition

Following the baseline work [2], the problem in CPCC is to optimise a trade-off between bitrate and classification accuracy, formalised as minimising:

$$\mathcal{L} = \mathcal{D}(\mathbf{t}, \mathbf{t}') + \lambda \mathcal{R}, \tag{4.1}$$

where $\mathcal{D}(\mathbf{t}, \mathbf{t}')$ is the cross-entropy between ground-truth $\mathbf{t}$ and predicted labels $\mathbf{t}'$, $\mathcal{R}$ the bitrate of quantized latents, and λ the balancer, echoing the Information Bottleneck [166]. Point clouds are represented as $\mathbf{X} \in \mathbb{R}^{3\times N}$, with the neural model $\mathcal{F}_\theta$ mapping to bitstream $\mathcal{B}$ for classification without full reconstruction.

4.3.2 Architecture of Our CPCC-PES

This work's architecture consists of an encoder layer, an entropy bottleneck layer, and a decoder layer. More details of the architecture are shown in Fig. 4.2.

As shown in Fig. 4.2, the input point cloud is converted to high-dimensional latent features through an embedding layer formed by Multilayer Perceptrons (MLPs).

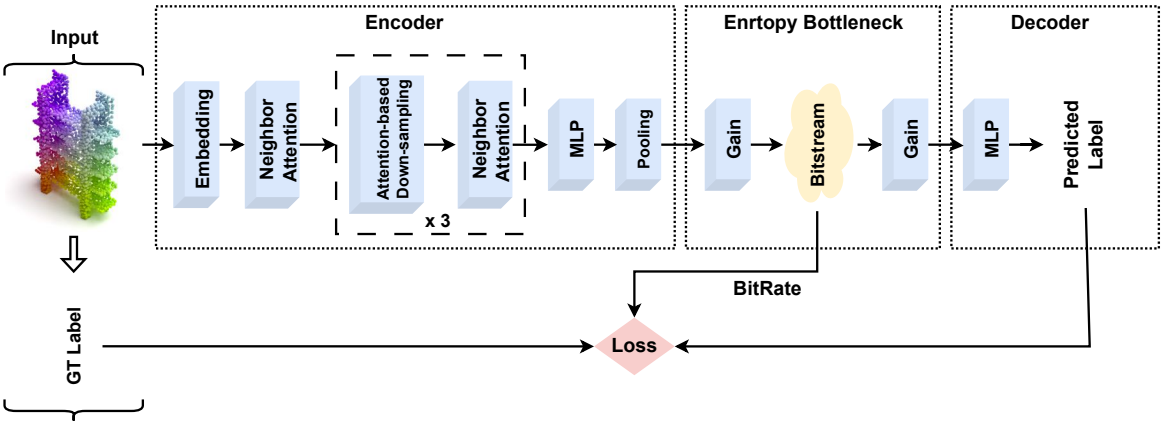


Figure 4.2 : This figure illustrates the architecture of the CPCC-PES network. The workflow begins with the Input labelled as GT Label, processed through Embedding, Neighbour Attention, three stages of Attention-based Down-sampling, a MLP, and pooling in the encoder to create the latent representation for the Entropy Bottleneck, generating the bitstream with associated bitrate. In the decoder, it applies Gain Layers and a MLP to produce the Predicted Label, optimised via Loss for end-to-end training. Further technical details are described in Sect. 4.3.2.

Then, the Neighbour Attention layer learns and transfers these latent features to the Attention-based edge sampling layer. This down-sampling layer selects the edge information and point cloud features based on local attention, which serves as an ideal normalised correlation map to measure their difference, providing a learnable adaptive approach for edge point and feature selection. Inspired by the Attention-based point cloud edge sampling (APES) work in [117], the normalised correlation map is computed by the attention map between the centre point and its neighbours to identify how much the features of surrounding points differ from the central point (see Eq. 4.2). A larger standard deviation in the normalised correlation map means a higher possibility that it is an edge point. This edge sampling method mitigates the feature-learning limitation of the encoder of LPCCC [2] and achieves a better trade-off between accuracy and bitrate.

Moreover, in contrast to APES [117] and inspired by advanced learning-based point cloud methods such as D-PCC [36], three down-sampling layers are utilised to better control bitrate, accompanied by a down-sampling ratio of 0.5. After each down-sampling layer, the edge points of the point cloud can be obtained. Then, the corresponding representative edge features will be selected and learned based on these edge points through a Neighbour Attention Layer. These learned features will concatenate together and pass through an MLP layer and a pooling layer to generate the final input permutation-invariant feature vector.

Then, the vector will pass through an Entropy Bottleneck layer to get the coded bit-stream. Finally, in the decoder, the decoded feature vector will pass through two gain layers and an MLP layer to increase its adaptive capability and obtain better performance to get the predicted label.

4.3.3 Attention-based Edge Sampling

The Canny edge detector [149] is widely used in edge detection for 2D images. When applying it, the intensity gradient of each pixel of the image, such as the strength of a colour change, is computed and then compared with other pixels' gradients within a local patch area. The pixels that have a larger gradient intensity difference are recognised to be the edge pixels, which can outline the edge details of the images.

In detail, a normalised correlation map is calculated for each pixel/point and its local neighbourhood, then the standard deviation σ in used to select edge candidates—higher σ indicates greater variability in correlations, akin to strong intensity gradients in traditional edge detection. For instance, as illustrated in Fig. 4.3, consider the standard deviations of potential edge pixels A (at a sharp colour boundary) and B (in a smoother area), denoted as σ_A and σ_B, and similarly for 3D points C (with diverse neighbour relations) and D (more uniform), as σ_C and σ_D. In practice,

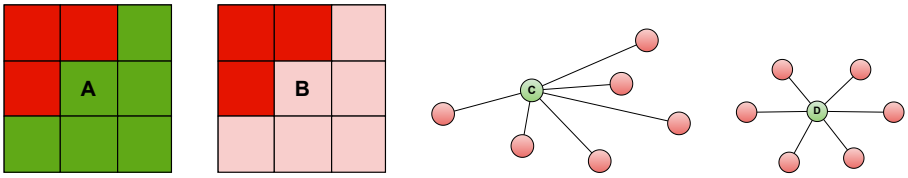


Figure 4.3 : Left: 2D pixel examples showing local neighbourhoods (grids) for potential edge pixel A (at a colour boundary, *(e.g.*, red-green interface) and non-edge pixel B (in a uniform area, *(e.g.*, red-pink transition). Right: 3D point cloud examples with central points C (likely on an edge, with diverse neighbour connections indicating variability) and D (non-edge, with more uniform connections). For each, a normalised correlation map is computed between the centre pixel/point and its neighbours (including itself). Edges are identified by comparing standard deviations (σ): higher σ signals greater variability, marking A and C as edge candidates. Further technical details are described in Sect. 4.3.3.

$\sigma_A > \sigma_B$ because A's neighbours span disparate regions (*e.g.*, red and green), leading to uneven correlations and higher variability; likewise, $\sigma_C > \sigma_D$ due to irregular connections in point clouds. Thus, pixels/points like A and C with elevated σ are prioritised as edges, extending Canny's gradient-based local comparison to irregular 3D data via attention maps.

Specifically, to extend the edge detection to irregular point cloud areas, we use K-Nearest Neighbour (KNN) to define a local patch, and we also compute a normalised correlation map to find and differentiate the edge points within a point patch. Inspired by [117], the attention map is an ideal option to serve as the normalised correlation map between point features within each patch.

Assuming that the input point cloud set has n points (where n denotes the total number of points), which can be represented as $\boldsymbol{P} = \rho_i{}_{i=1}^{n} \in \mathbb{R}^{n\times 3}$ (with ρ_i denoting the 3D coordinates of the i-th point). For each point ρ_i, its neighbour

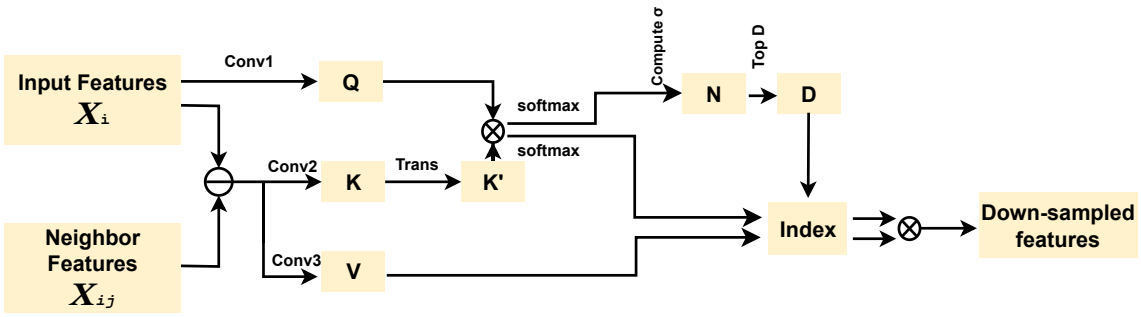


Figure 4.4 : This figure illustrates the architecture of Local-Attention-based Edge Sampling. The workflow starts with Input Features processed via Conv1 to generate the query matrix Q, while Neighbour Features are handled by Conv2 to produce the key matrix K (with transpose) and Conv3 to produce the value matrix V; Q and K are multiplied, normalised via Softmax σ_i, and selecting D points from the N points based on the top σ_i values and finally obtaining the down-sampled features. Further technical details are described in Sect. 5.3

point ρ_j (where j indexes the neighbours) is found by KNN (with k as the number of neighbours) from a point patch. Thus, as shown in Fig. 4.4, within a patch, the input features for the centre point ρ_i can be represented as X_i (the feature vector of ρ_i), the corresponding features of the neighbour point ρ_j are X_{ij} (the feature vector of ρ_j relative to ρ_i), and the feature difference between the centre point ρ_i and its neighbour ρ_j can be represented as $X_{ij} - X_i$. Then, X_i will be sent to $conv1$ (a convolutional or fully connected layer for query projection) to get the query Q (the query matrix). Meanwhile, $X_{ij} - X_i$ will be sent to $conv2$ (a convolutional or fully connected layer for key projection) and $conv3$ (a convolutional or fully connected layer for value projection), to obtain the key K (the key matrix) and the value V (the value matrix), respectively. Following the design of Attention mechanism [70], the attention map can be computed as QK^T (the dot-product similarity scores).

In this context, the local attention correlation map can be calculated as:

$$\mu(X_i, X_{ij}) = \psi(X_i)^T \varphi(X_{ij} - X_i). \tag{4.2}$$

Here, ψ and φ are fully connected layers where inputs are the centre point features X_i and the feature difference between the neighbour point and the centre point, $X_{ij} - X_i$, respectively. $\psi(X_i)$ is the query Q and $\varphi(X_{ij} - X_i)$ is the key K.

Because calculating the attention also involves $softmax$ normalisation and scaling with a factor $\sqrt{d}$, the final equation of the normalised correlation map $\boldsymbol{M}_i$ regarding point ρ_i can be written as:

$$\boldsymbol{M}_i = softmax\left(\mu(X_i, X_{ij})/\sqrt{d})\right). \tag{4.3}$$

After the correlation map $\boldsymbol{M}_i$ is obtained, the standard deviation σ_i of the map $\boldsymbol{M}_i$ can be calculated. This work can then select the edge points' indexes and their corresponding features by selecting those points with a higher σ_i. For example, in Fig. 4.4, this work obtains the down-sampled index by selecting D points from the N points based on the top σ_i values and then obtains the down-sampled features.

4.3.4 Neighbour Attention Aggregation

After each attention-based down-sampling layer, this work can obtain the down-sampled point cloud edge points and their corresponding features. Considering that the down-sampling ratio is set to 0.5, the dimensions of the down-sampled point features will be 1024, 512, and 256 at each stage in turn. To further increase the performance of CPCC-PES, this work decided to incorporate an attention mechanism [70] here to learn and aggregate the obtained down-sampled point features. It makes use of the attention mechanism from APES [117] as a stronger tool to capture the features that are focused on the edge information of the point cloud.

This work first finds the centre point (denoted as ρ_i, where ρ_i represents the coordinates of the i-th point) and its k corresponding neighbour points (denoted as ρ_j, where j indexes the neighbours) by using the KNN and set k to 32 (where k is the number of neighbours). As shown in Fig. 4.5, the features of the centre point are set as the Input Features (denoted as X_i, the feature vector of ρ_i) and are passed through a $conv1$ layer to get Q (the query matrix), and the feature difference between the centre point and its neighbours (denoted as $X_{ij} - X_i$, where X_{ij} is the feature vector of neighbour ρ_j) will pass through $conv2$ and $conv3$ separately to get K (the key matrix) and V (the value matrix). Then, the equation of the neighbour attention features $\boldsymbol{F}_N$ (the aggregated attention features from neighbours) can be represented as follows:

$$\boldsymbol{F}_N = softmax\left((QK^T)/\sqrt{d})\right)V, \tag{4.4}$$

where $\sqrt{d}$ is the scaling factor (with d denoting the feature dimension). Finally, the output feature will be the concatenation between the neighbour attention features $\boldsymbol{F}_N$ and the input features (*i.e.*, X_i).

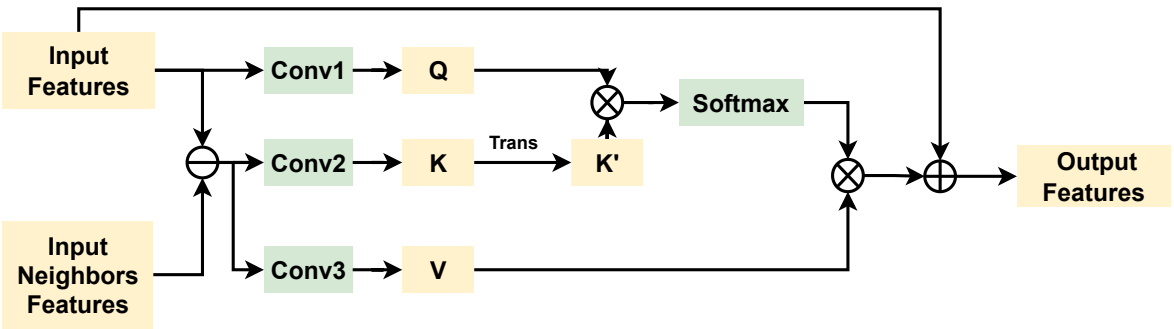


Figure 4.5 : This figure shows the information of Neighbour Attention Aggregation. The features of the centre point are set as the Input Features to get Query Q, Key K, and Value V. Then, the attention will aggregate features from both the Input and the Neighbours to get the output Features.

4.3.5 Entropy Bottleneck

Gain Layer: After max-pooling and normalisation in the encoder, the result vectors are composed of all small values, which is not friendly to the following entropy calculation, and also the classifiers in the decoder. To increase the adaptability and stability of the Entropy Bottleneck, this work follows the ideas from LPCCC [2] to add two Gain layers. Each Gain layer is defined to be formed with a trainable vector, and the elements in this trainable vector are initialised with integers. In the experiment, this work sets the initial values of each of the elements of this trainable vector to 10. Then, these trainable vectors will be multiplied element-wise with the input feature vectors to amplify the small values and differences existing in the input.

Entropy Model: The first gain layer will receive the learned feature vectors provided by the Encoder and output an adaptive version of the feature vector. To further obtain the bit-streams, this work uses integer rounding to quantify these vectors. During training, uniform quantisation is simulated using additive uniform noise $\mu \in (-0.5, 0.5)$. Then, the quantized vector is losslessly encoded using a fully factorised learned entropy model [167].

4.3.6 Decoder

To reduce the overall computational complexity this work follows the work of LPCCC [2] and designs a lightweight decoder module. In the decoder, this work uses a lightweight MLP classification layer that turns the initial decoded bit-stream into predicted labels.

4.3.7 Loss Function

In the Point Cloud Compression field, researchers use a trade-off between bitrate and distortion loss, which is measured by Chamfer Distance to quantify the

performance of the compression model [60]. Similarly, in CPCC, a trade-off between bitrate and classification accuracy can also be considered [2].

Following the idea of Information Bottleneck (IB) [166], LPCCC [2] has given evidence that the below equation, as a trade-off between bitrate and accuracy, is analogous to the IB, and it is suitable to be the loss function in CPCC:

$$L = R + \lambda \cdot D(t, \hat{t}). \tag{4.5}$$

Here, L represents the total loss value, R represents the bitrate value estimated by the entropy model, λ represents the penalty weight, and $D(t, \hat{t})$ represents the cross-entropy loss between one-hot encoded ground truth label t and the softmax of the prediction label $\hat{t}$ obtained by the model. This work also utilises this method to calculate the total loss.

4.4 Experimental Results and Discussion

4.4.1 Dataset and Implementation Details

To rigorously evaluate classification performance, this work utilises the ModelNet40 [6] dataset, which contains 12,311 manufactured 3D CAD models spanning 40 common object categories. ModelNet40 is specifically selected because it serves as the universally acknowledged standard benchmark for isolated 3D object classification algorithms (*e.g.*, PointNet [42]) and existing compressed-domain baselines like LPCCC [2], ensuring direct and fair comparability.

While evaluating the proposed framework on large-scale autonomous driving datasets (such as KITTI [5] or nuScenes [111]) would be highly relevant for real-world vehicular applications, those datasets are primarily tailored for complex scene-level tasks (*e.g.*, 3D object detection or semantic segmentation) rather than isolated object-level classification. Similarly, while ShapeNet is excellent for dense geometry reconstruction (as explored in Chapter 3), ModelNet40 [6] remains the de facto

standard for validating end-to-end classification architectures. Therefore, aligning with current CPCC literature, this chapter focuses strictly on optimising object-level classification, identifying scene-level compressed-domain detection evaluation as a critical direction for future work.

The dataset [6] is split into training with 9,843 models and testing with 2,468 models. For each model, this work uniformly samples points from the mesh surface and normalises them to the unit sphere. The input only contains point cloud coordinate information, and no data augmentation methods are used.

This experiment was trained and tested on a computer with an i9-13900K 3.61GHz CPU, 24 GB RAM, and an NVIDIA GeForce RTX 4090 GPU. During training, this work sets the penalty weight λ in the loss function (see Eq. 4.5), which ranges from 20 to 16,000, the initial learning rate was 0.01, the batch size of 8, and Adam optimisation.

4.4.2 Performance Comparison and Evaluation

As a common evaluation metric, Top-1 Accuracy has been widely used in many CPCC works, such as VM-CPCC [155] and LPCCC [2]. For experimental comparison, because VM-CPCC is a voxel-based CPCC method, this work compares its results with point-based LPCCC [2] in a compressed way and other baseline works, including G-PCC TMC13 [50], OctAttention [39], and IPDAE [138] in a decompressed way, as shown in Table 4.1.

In addition to the conventional metric BD-Rate [49], this work also proposes an evaluation metric called BD-Top-1 to evaluate the different models' power in achieving less bitrate for higher accuracy. This new metric is crucial because traditional BD-Rate focuses on bitrate savings at equivalent quality levels (*e.g.*, PSNR), but in CPCC contexts where "quality" is task-specific classification accuracy, a complementary measure is needed to quantify average accuracy improvements over

a bitrate range. By adapting the Bjøntegaard Delta framework from video compression, BD-Top-1 provides a single, interpretable value that captures the integral difference in accuracy curves, enabling a more comprehensive assessment of how efficiently a model delivers superior performance at reduced bitrates—addressing a gap in existing evaluations that often rely solely on point-wise comparisons or visual curves.

Top-1 Classification Results on ModelNet40:

To demonstrate the efficiency of the proposed CPCC-PES, this work compares the proposed method with baseline DPCC methods. In addition, to highlight and compare the differences in bitrate across various CPCC and DPCC techniques, this work compares CPCC-PES against the baseline method PointNet [42], which is commonly employed as a baseline in many DPCC tasks, such as the transformer-based APES [117] and LPCCC [2]. This work applies these methods to the same point cloud dataset and utilises the same input for encoding and decoding. Then, following the practice in LPCCC [2], this work employs the PointNet [42] classifier to obtain the final classification result. Fig. 4.6 illustrates the comparative results.

From Fig. 4.6, this work shows that the proposed codec has superior performance compared with the baseline CPCC and DPCC methods, demonstrating that our codec can achieve higher classification at lower bitrates. In particular, considering the baseline PointNet [42]'s Top-1 Accuracy is 89.2%, this work achieves the baseline by only consuming about 0.06 bpp (bits per point) and 0.07 bpp. This work can also achieve 90.8% Top-1 Accuracy with only 0.2 bpp cost, while LPCCC [2] needs to spend 0.45 bpp to obtain a Top-1 Accuracy of 88.5%. It is worth noting that PointNet [42] proposed input geometric transformations to align point clouds for better classification performance and reported their results with and without the affine transformations. Since the proposed compression model was trained without

the input spatial alignment, the baseline without geometric transformations offers a fairer comparison.

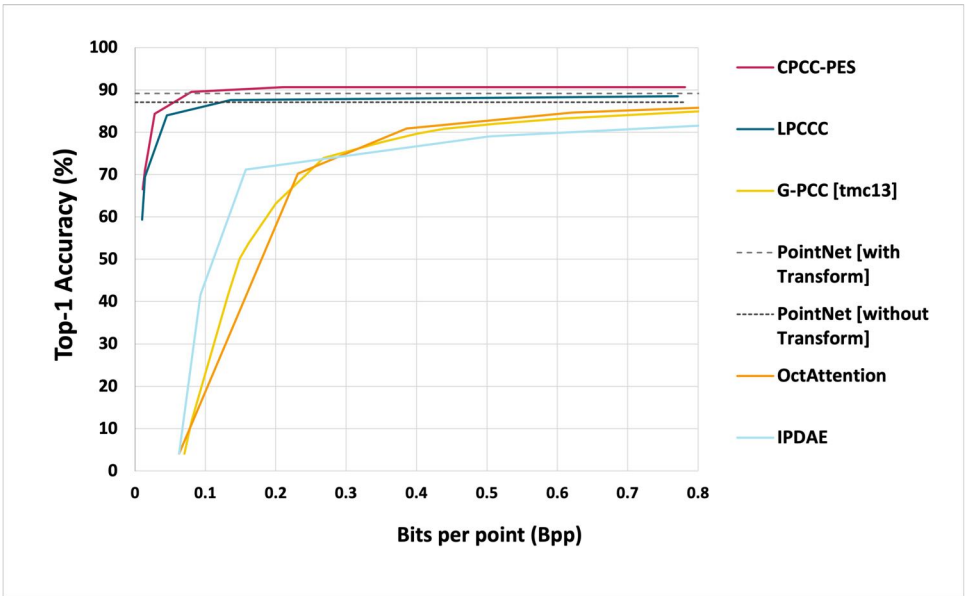


Figure 4.6 : Top-1 classification results on ModelNet40 at different Bpp rates.

BD-Rate and BD-Top-1 Accuracy: The video standardisation community has used Chamfer Distance and BD-PSNR [121] [168] measurements for many years as a method for evaluating the performance of new codecs and video coding tools. In particular, BD-PSNR can measure the average rate difference between two PSNR rate-distortion curves and can be calculated as:

$$BD\text{-}PSNR = \frac{1}{R_1 - R_2} \int_{R_1}^{R_2} [PSNR_1(R) - PSNR_2(R)]\, dR \tag{4.6}$$

where R_1 and R_2 are the distortion limits, and $PSNR_1(R)$ and $PSNR_2(R)$ are the PSNR values of two codecs at bitrate R.

Following the metric of BD-PSNR, this work proposes the BD-Top-1 Accuracy metric, denoted as $BD\text{-}Top\text{-}1Accu$, to evaluate the performance of different CPCC codecs in terms of their bitrate efficiency while achieving high Top-1 Accuracy. BD-Top-1 Accuracy quantitatively provides a comprehensive insight into the codec's

ability to maintain high point cloud classification accuracy at minimised bitrate levels. Similar to the design of BD-PSNR, our BD-Top-1 Accuracy metric is defined as:

$$BD\text{-}Top\text{-}1Accu = \frac{1}{R_1 - R_2} \int_{R_1}^{R_2} [Top\text{-}1Accu_1(R) - Top\text{-}1Accu_2(R)] \, dR \quad (4.7)$$

where $BD\text{-}Top\text{-}1$ Accu is the BD-Top-1 Accuracy, and the $Top\text{-}1$ $\text{Accu}_1(R)$, $Top\text{-}1$ $\text{Accu}_2(R)$ are the Top-1 Accuracy values of two codecs at bitrate R.

Table 4.1 shows the comparison of BD metrics with the maximum attainable accuracy with baseline methods. Following [49], in evaluation metrics, BD-Rate measures the relative bitrate value difference between two CPCC performances. A CPCC method with a lower BD-Rate will spend less bitrate while achieving the same point cloud classification accuracy. In general, our model can save over 24% bitrate compared to the baseline methods. This work also saves over 95% bitrate to achieve the same accuracy performance compared with the DPCC methods.

Table 4.1 : Comparison of BD-Rate and BD-Top-1 accuracy with recent baseline codecs.

Compared with CPCC method	BD-Rate	BD-Top-1 Accuracy
CPCC-PES vs LPCCC [2]	-24.24%	-3.02%
Compared with DPCC methods	**BD-Rate**	**BD-Top-1 Accuracy**
CPCC-PES vs IPDAE [138]	-96.12%	-22.98%
CPCC-PES vs OctAttention [39]	-94.54%	-30.2%
CPCC-PES vs G-PCC [50]	-95.2%	-15.2%

Discussion on Evaluation Symmetry: It is crucial to explicitly acknowledge an inherent asymmetry in this comparative methodology: the proposed CPCC-PES and the baseline LPCCC [2] are end-to-end task-driven architectures trained

specifically to optimise classification accuracy (via cross-entropy loss) directly in the compressed domain. Conversely, the generic DPCC methods (*e.g.*, G-PCC [31], OctAttention [39], IPDAE [138]) are fundamentally optimised for geometric reconstruction fidelity. The decompressed point clouds from these generic codecs are subsequently fed into a standard, pre-trained PointNet [42] classifier that has not been explicitly fine-tuned on their specific compression artifacts.

While retraining the downstream classifier for each individual codec's distinct artifacts might yield marginally higher accuracies for the DPCC methods, the current evaluation reflects the standard practical deployment paradigm, where off-the-shelf pre-trained analysis networks are routinely applied to decompressed data. This comparison does not strictly evaluate generic compression algorithmic superiority. Rather, it validly demonstrates the substantial rate-accuracy efficiency gains achievable by adopting a task-specific compressed-domain architecture over the traditional "compress-then-analyse" pipeline for resource-constrained edge applications.

4.4.3 Ablation Studies

This section investigates the potential of the CPCC-PES approach for smaller datasets in CPCC tasks by comparing the results with the baseline work.

Performance of CPCC-PES with Small Numbers of Points: Typically, each point cloud model contains 2048 points, randomly sampled from the ModelNet40 [6] dataset. To assess the adaptability of the proposed CPCC-PES to smaller point cloud models, this work reduces the input point cloud size using a random sampling technique to 1024, 512, and 256 points.

Table 4.2 compares the Max Top-1 Accuracy results of our method and those of the baseline LPCCC [2] work under different numbers of input points. Max Top-1 Accuracy refers to the maximum Top-1 Accuracy that the method can achieve in point cloud classification without considering the bitrate consumption.

The result in the table shows that the proposed method consistently outperforms LPCCC [2] in Max Top-1 Accuracy even with reduced point cloud size. This shows that the proposed CPCC method is robust to different sizes of point clouds. Specifically, this work achieves about 2% higher Max Top-1 Accuracy compared to the baseline work LPCCC [2].

Table 4.2 : Comparison of Max Top-1 Accuracy with the baseline work LPCCC [2] concerning various numbers of input points.

Number of input points	CPCC-PES Max Top-1 Accuracy	LPCCC [2] Max Top-1 Accuracy
1024	**90.8%**	88.5%
512	**90.35%**	88.0%
256	**90.02%**	87.6%

4.4.4 Edge Sampling Visualisation

Fig. 4.7 below visualises the local-attention-based edge sampling point cloud models obtained from ModelNet40 [6]. From top to bottom, the point clouds displayed represent *television*, *plane*, *desk*, and *bed*. From left to right, each model contains 2048, 1024, 512, and 256 points, respectively. It can be seen from the figure that, by using the local-attention-based attention map, this work can better represent the primary structural-outlier edge information of the point cloud.

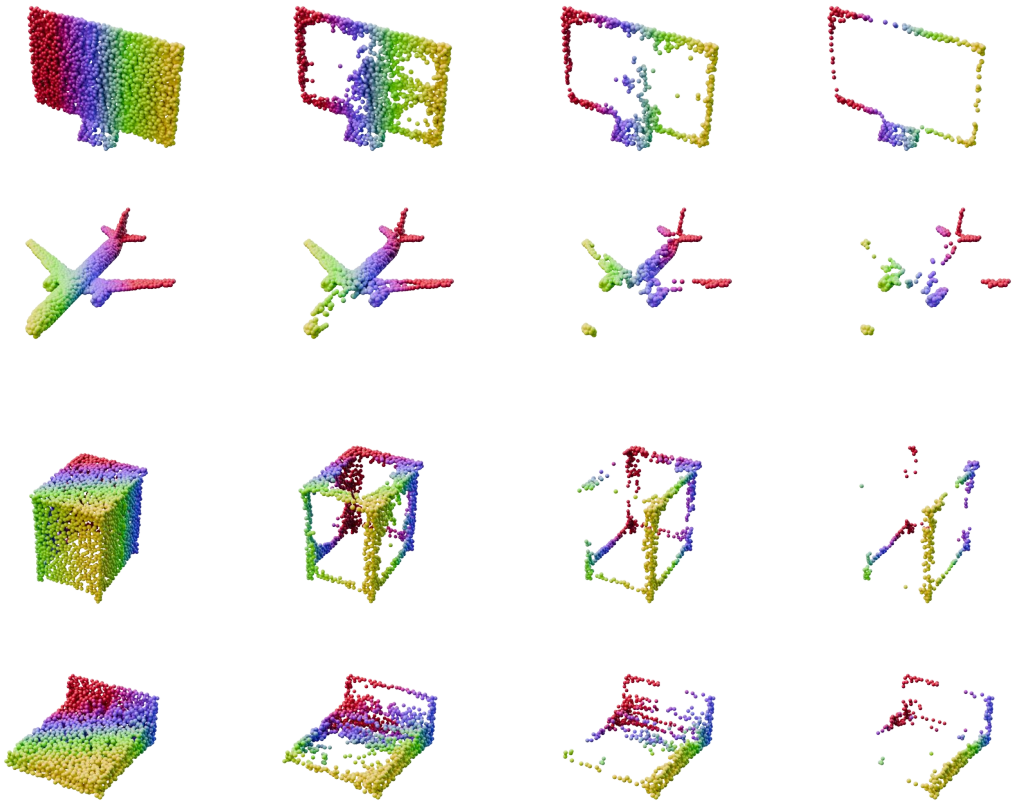


Figure 4.7 : A visualisation of the proposed method result. From top to bottom, the point clouds are *television*, *plane*, *desk*, and *bed*. From left to right, each point cloud model contains 2048, 1024, 512, and 256 points, respectively.

4.5 Summary

This section proposes a new approach for point cloud classification in the compressed domain by leveraging attention-based point cloud edge sampling. By extending the success of edge detection to point clouds in an effective way, this approach enhances the classification accuracy and performance at a reduced bitrate. These experiments demonstrate that this method has a BD rate reduction of over 24% compared with baseline methods. For future work, there are several promising directions for further exploration. These include exploring end-to-end compressed point cloud segmentation, evaluating the architecture on large-scale autonomous

driving datasets such as KITTI, and developing more efficient classifiers. Furthermore, a recognised limitation of current CPCC evaluations is the inherent asymmetry when comparing task-specific models against generic DPCC pipelines without artifact-specific classifier retraining. Future work should investigate joint-training strategies that simultaneously optimise generic compression codecs and downstream analysis networks, aiming to establish a rigorously unified benchmark for evaluating rate-accuracy trade-offs across diverse compression paradigms.

Chapter 5

Scalable Point Cloud Compression with Density-aware Tail-drop

Building on the local feature preservation from Chapter 3 and compressed-domain efficiency from Chapter 4, while addressing the scalable coding gaps in single-rate methods noted in Chapter 2's survey, this chapter presents ScalaDAT-a new density-aware tail-drop framework for scalable point cloud compression (SPCC) that enables scalable reconstruction through a single training iteration. This chapter begins with an introduction to point cloud coding challenges, traditional and learning-based methods, and the pivotal gap in scalable geometry coding for large-scale datasets with varying densities, while drawing cross-influences from established 2D image and video scalable paradigms to motivate this chapter's approach and outline key contributions. Following this, this chapter introduces the ScalaDAT architecture and density-aware tail drop module, inspired by stochastic dropping techniques, preserving high-density region information. Then, it shows the experimental results and conclusions. Experimental validation on diverse benchmarks demonstrates superior BD-Rate gains (over 28.6% on SemanticKITTI and 18.15% on ShapeNet relative to baseline methods). Finally, results are interpreted through ablations and visualisations, concluding with insights on future scalable PCC for real world applications.

5.1 Introduction

Point clouds, captured by technologies like LiDAR scanners, deliver rich 3D environmental representations through millions of points with attributes such as colour and intensity. Their irregular, unstructured nature—unlike grid-based 2D im-

ages—poses profound challenges for storage, transmission, and processing [3]. With single scans yielding hundreds of millions of points, PCC becomes indispensable for applications in autonomous driving, VR/AR, and 3D mapping [33, 169, 14, 15], demanding techniques that maintain geometric fidelity amid irregularity. Traditional methods like G-PCC [31] and V-PCC [32] hierarchically encode geometry or project it onto 2D planes for video codec leverage [3]. Yet, on large-scale datasets such as SemanticKITTI [5], they suffer from inefficiency, quality degradation, and computational burdens. Learning-based PCC, conversely, employs deep networks to grasp spatial intricacies [1, 60, 37], inspired by hyperprior models [167], excelling at comparably low bitrates but often generating monolithic bitstreams (Fig. 5.1), which is unsuitable for bandwidth-variable scenarios.

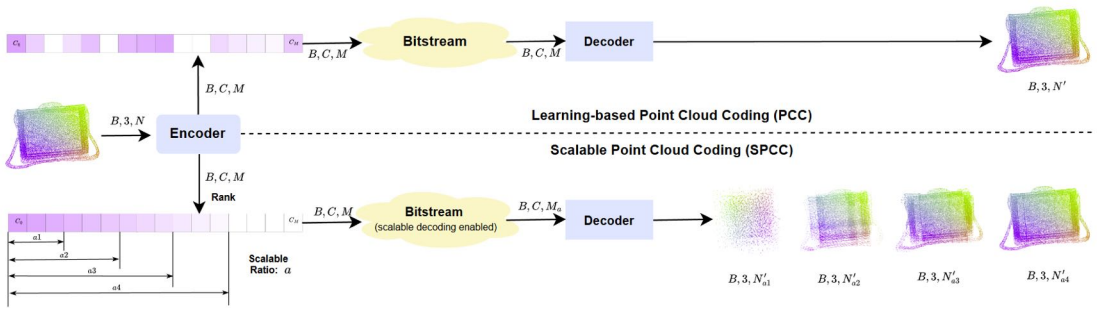


Figure 5.1 : Comparisons between Point Cloud Coding (PCC) and Scalable Point Cloud Coding (SPCC). SPCC generates a scalable bitstream post-encoding, enabling customised decoding for variable coding ratios a and reconstruction qualities.

As conceptually illustrated in Fig. 5.1, Scalable Point Cloud Compression (SPCC) enables partial decoding for incremental refinement, allowing scalable reconstructions via a Scalable Ratio (SR) α, which dynamically trades bitrate for geometric quality. It is crucial to formally distinguish the specific type of scalability addressed here. Traditional standard codecs, such as G-PCC [31], inherently possess spatial scalable capabilities (often referred to as scalable decoding) through their hierarchical octree Level-of-Detail (LoD) subdivisions. Similarly, foun-

dational learning-based multiscale frameworks like SparsePCGC [52] achieve scalability across different spatial resolutions via sequential refinement stages.

However, achieving fine-grained, *channel-based* scalable geometry coding utilizing a single-training paradigm, where scalability is achieved by dynamically pruning latent feature channels rather than relying on discrete spatial octrees or multi-stage resolutions, remains an active area of research. Recent learning-based advancements focus primarily on attribute compression, as demonstrated by Rudolph *et al.* [54]. To bridge the gap in geometry coding, this work draws inspiration from established channel/layer scalable coding paradigms in 2D videos (e.g., H.264 SVC [30]), enabling efficient data delivery and dynamic bandwidth adaptability in constrained environments without claiming strict low-latency guarantees.

Then, ScalaDAT is proposed, which adapts the cross-influences from 2D domains—such as selective feature prioritisation [105] and multi-rate layering [170]—by extending D-PCC [36] with variance-gradient channel importance and density-aware tail-drop, further informed by stochastic scalable techniques [170, 105]. By leveraging density as a proxy for local complexity [45, 36], this approach addresses the irregular, unstructured geometry of point clouds. Density, a pivotal factor in point cloud processing [45] [36], serves as an indicator of local detail complexity. High-density regions typically indicate intricate geometry (*e.g.*, object surfaces), whereas sparse areas usually correspond to simpler, less critical structures. By exploiting density variations, ScalaDAT prioritises structurally significant details, improving rate-distortion efficiency through informed feature selection. In detail, this approach delivers the following key contributions:

- We propose ScalaDAT, a new framework that achieves scalable point cloud coding with a single training stage.
- We develop a density-aware tail-drop approach that leverages point and struc-

ture density to prioritise structurally significant regions, enhancing efficiency by focusing on complex, high-density areas.

- We validate ScalaDAT on SemanticKITTI and ShapeNet, demonstrating superior BD-Rate performance compared to baseline methods.

5.2 Literature Review

5.2.1 Point Cloud Coding

Traditional point cloud coding techniques employ spatial data structures and quantisation to reduce redundancy without neural networks. For example, MPEG's G-PCC [31] uses octree representations and triangular surface coding for geometric encoding in static scenes, while V-PCC [32] projects 3D data onto 2D planes to exploit video codecs for temporal redundancy in dynamic sequences. However, these methods incur high computational complexity, substantial storage needs, and limited scalability across datasets. To mitigate these limitations, learning-based methods leverage deep neural networks for compact, efficient representations, surpassing traditional approaches in scalability and performance.

Point-based methods process raw points directly, avoiding losses from intermediate transformations. Inspired by PointNet++ [41], Hao *et al.*'s KNN-based encoding [80] and D-PCC's density preservation [36] enhance reconstruction quality and adaptability across sparse or irregular distributions. Octree-based methods refine hierarchical tree coding with neural networks. OctSqueeze [40] introduces probabilistic occupancy modelling, followed by Tingyu *et al.*'s hierarchical latent variables [83]. Attention mechanisms in OctAttention [39] further optimise context modelling, balancing efficiency and detail preservation. Voxel-based methods treat point clouds as occupancy grids, applying 3D CNNs. Quach *et al.*'s convolutional autoencoder [26], PCGC [60], and JPEG Pleno [1] pioneered this approach, while

sparse convolutions [37] and block-based partitioning [51] address sparsity inefficiencies. These learning-based strategies outperform traditional techniques, offering robust adaptability to diverse point cloud densities and structures with enhanced efficiency and reconstruction fidelity.

5.2.2 Scalable Image Coding

Scalable coding has been extensively explored in learning-based image coding, building on foundational works by Ballé *et al.* [171, 167] and Minnen *et al.* [172]. Early efforts partitioned latent representations into a base layer and enhancement layers, each decoded by separate networks to produce intermediate image versions [173]. Others employed Recurrent Neural Networks (RNNs) to scalably encode quantisation residuals [174, 175, 176], but these approaches suffer from high computational costs, memory usage, and low throughput [177].

Alternatively, Lu *et al.* [178] proposed nested quantisation, defining multiple levels with nested grids to scalably refine all latents from coarsest to finest. Building on this, Lee *et al.* [127] represented encoded features in ternary digits (trits) and transmitted them in decreasing significance order using rate-distortion priorities, sending critical information first. However, this results in degraded performance and degraded quality at low bitrates, often requiring a post-processing network for refinement. Similarly, Li *et al.* [179] replaced uniform quantisers with dead-zone quantisers to reduce redundant symbols, thereby improving the efficiency of scalable coding. Despite strong scalable coding performance in images, these methods [127, 179] rely on fixed-rate models, limiting their flexibility.

To overcome the limitations of RNN and nested quantisation, tail-drop techniques enable variable-rate coding, reducing computational complexity while improving coding efficiency. Koike *et al.* [170] showed that discarding the least important principal components yields variable rate dimensionality reduction with

graceful degradation. Based on this, Hojjat *et al.* [105] introduced a double-tail-drop scalable training protocol that prioritises latent and hyper-latent channels by relevance [171, 167], enabling transmission in order of channel importance and eliminating the need for nested quantisation.

5.2.3 Scalable Point Cloud Coding

Although scalable coding is well established in 2D images, its application to point clouds remains underexplored, which leads to a research gap. Notable advancements include Huang *et al.*'s [62] octree-based subdivision, which estimates geometric centres of tree-front cells to support scalable coding.

Recent work applies deep learning to do scalable point cloud coding. Rudolph *et al.* [54] employed quantisation residuals from prior representations with a learned lightweight transformation in the entropy bottleneck to enable scalable attribute coding. Similarly, Gokulnath *et al.* [180] projected the 3D model from multiple viewpoints to form a sequence of view-specific six-dimensional (RGB+XYZ) images on 2D grids, then used a symmetry-based convolutional neural pyramid to encode them scalably from coarse to fine, exploiting inter-projection redundancies for efficient transmission. Nonetheless, scalable geometry coding for point clouds remains underexplored, particularly on challenging benchmarks like SemanticKITTI [5] and ShapeNet [6]. The vast scale of LiDAR data and the geometric diversity of models pose significant challenges for PCC in both data volume and detail preservation. These challenges underscore the significance of scalable PCC for improving data efficiency in resource-intensive domains such as autonomous navigation and immersive media [54, 180].

To address this need, this work proposes ScalaDAT, which leverages density information to guide scalable coding and prioritise critical regions of the point cloud. Through tail-drop, it selectively preserves essential features, reducing decoder com-

putational load while enabling efficient, controllable scalable decoding.

5.3 Methodology

5.3.1 Problem Definition

Following the foundational point cloud representations and lossy compression paradigms formally defined in Equations 1.1 and 1.2 of Chapter 1, standard learning-based point cloud codecs optimise a fixed rate-distortion trade-off, generating a monolithic bitstream requiring full decoding.

In contrast, SPCC structures latents for incremental reconstruction from a single bitstream. In detail, ScalaDAT achieves SPCC by ranking latent channels by variance-based importance, enabling rate-scalable decoding through selective channel activation while preserving rate-distortion efficiency across different bitrates. To better demonstrate the scalable coding's capability and performance, we introduce a controllable Scalable Ratio (SR) $\alpha \in [0, 1]$ specifying the fraction of latent channels to activate in decompression, enabling rate-scalable reconstruction:

$$\mathcal{F}_{\theta}^{\text{scal}} : (\mathbf{X}) \mapsto \mathcal{B} \mapsto \mathcal{B}_{\alpha} \mapsto (\mathbf{X}'_{\alpha}), \tag{5.1}$$

where $\mathcal{B}_{\alpha} \subset \mathcal{B}$ represents a subset of the total bitstream containing approximately α proportion of the total bits. In decompression, the reconstruction quality monotonically improves with α, while computational cost and bitrate scale proportionally. This enables dynamic adaptation, allowing decoders to begin with minimal channels for coarse reconstruction and scalably refine quality by incorporating additional channels as bandwidth permits.

5.3.2 The ScalaDAT Framework

This framework extends the density-preserving D-PCC architecture [36] by transforming a monolithic coding model into a scalable one with density-aware adaptation

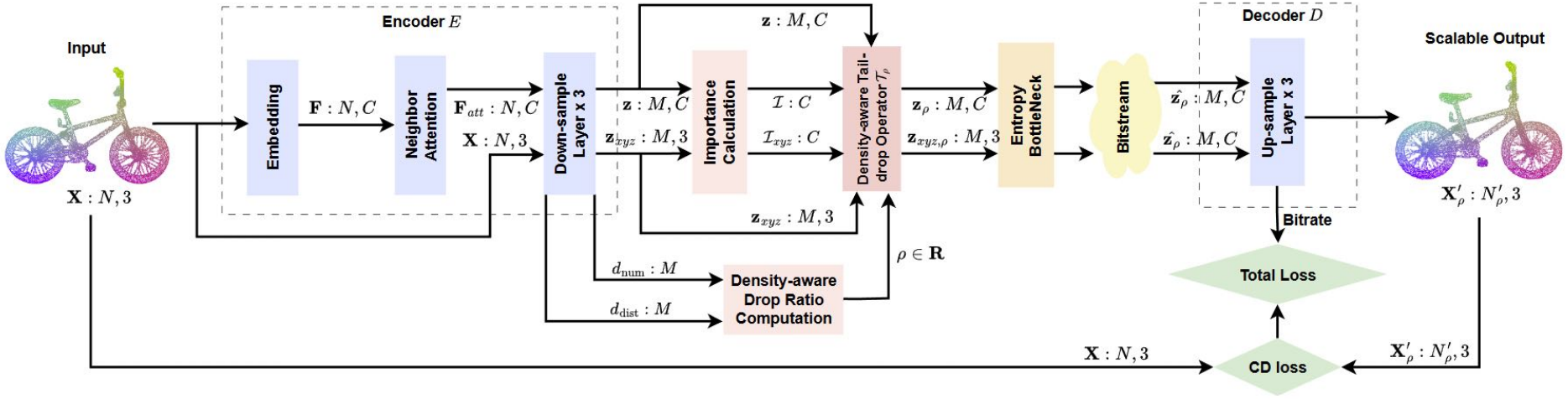


Figure 5.2 : This figure illustrates the architecture of ScalaDAT. This scalable coding model for point clouds leverages an end-to-end autoencoder with specialized blocks, including an Encoder E, a Density-aware Tail-drop Operator, an Entropy Bottleneck, and a Decoder D. The total loss is formed by a trade-off between density loss, Chamfer Distance (CD) loss, and Bitrate. Further technical details will be described in Sect. 5.3.

capabilities. As shown in Fig. 5.2, the overall design follows an autoencoder architecture specialised for point cloud processing and entropy-constrained coding. The encoder consists of three downsampling stages, configurable with factors of 1/2, 1/3, or 1/4, and a decoder with a maximum upsampling factor of 8. The scalable coding process is formalised as:

$$\begin{aligned} \mathbf{F} &= F(\mathbf{X}), \\ \mathbf{z}, \mathbf{z}_{xyz}, \mathbf{d} &= E(\mathbf{X}_{xyz}, \mathbf{F}), \\ \mathbf{z}_\rho, \mathbf{z}_{xyz,\rho} &= \mathcal{T}_\rho(\mathbf{z}, \mathbf{z}_{xyz}, \mathbf{d}), \\ \hat{\mathbf{z}_\rho}, \hat{\mathbf{z}}_{xyz,\rho} &= \mathcal{B}_z(\mathbf{z}_\rho), \mathcal{B}_{xyz}(\mathbf{z}_{xyz,\rho}), \\ R_\rho &= \mathrm{BPP}(\hat{\mathbf{z}_\rho}, \hat{\mathbf{z}}_{xyz,\rho}), \\ \mathbf{X}'_\rho &= D(\hat{\mathbf{z}_\rho}, \hat{\mathbf{z}}_{xyz,\rho}). \end{aligned} \tag{5.2}$$

Here, F extracts initial features $\mathbf{F} \in \mathbb{R}^{C\times N}$, which are refined by the encoder E through a series of down-sampling layers equipped with a Transformer mechanism. This yields downsampled coordinates $\mathbf{z}_{xyz} \in \mathbb{R}^{3\times M}$, latent features $\mathbf{z} \in \mathbb{R}^{C\times M}$, and density statistics $\mathbf{d} \in \mathbb{R}^M$. Here, M denotes the number of down-sampled points and C the feature channels. The density-aware tail-drop operator $\mathcal{T}\rho$ prunes latent representations according to a drop ratio $\rho \in [0, 1]$ (linked to the SR $\alpha = 1 - \rho$). Quantization is performed by Entropy bottlenecks $\mathcal{B}_\mathbf{z}$ and $\mathcal{B}_{\mathbf{z}_{xyz}}$, from which the bitrate R_ρ is computed. The decoder D reconstructs $\mathbf{X}'_\rho \approx \mathbf{X}$, with reconstruction quality directly controlled by ρ.

The density-aware tail-drop mechanism is pivotal for enabling scalable coding. Training employs a stochastic drop ratio ρ as in [170, 105], which is dynamically adjusted based on local density statistics [36]. This allows the model to prioritise dense regions while adaptively handling sparse areas, making it robust to incomplete feature representations. This stochastic training strategy allows the model to reconstruct point clouds from varying levels of latent feature completeness during

scalable coding at inference based on customised $\rho \in [0, 1]$. In inference, varying ρ produces reconstructions at different bitrates without retraining, enabling efficient scalable coding with a single trained model.

5.3.3 Channel-Importance based Density-aware Tail Dropping

5.3.3.1 Density-Aware Tail Dropping

In contrast to uniform channel drop techniques in all regions, such as those implemented in ProgDTD [105] and [170], this approach introduces a density-aware tail-drop mechanism with the purpose of adapting to the local density characteristics of the point cloud. Specifically, for point cloud $\mathbf{P} = \{p_1, ..., p_N\}$ and its downsampled set $\mathbf{X} = \{x_1, ..., x_M\}$, the density-aware drop ratio, denoted by ρ, is defined as:

$$\rho = \rho_{\max} - (\rho_{\max} - \rho_{\min}) \cdot \delta, \tag{5.3}$$

where $\rho_{\min}$ and $\rho_{\max}$ define the range of the drop ratio, empirically set to 0.15 and 0.4, and δ is the composite density score, governing channel preservation during coding, which is calculated by averaging normalised point concentration with inverted normalised distance as:

$$\delta = \frac{1}{2}\left(\frac{d_{\text{num}}}{d_{\max}} + \left(1 - \frac{d_{\text{dist}}}{m_{\max}}\right)\right). \tag{5.4}$$

Here, d_{num} quantifies local point concentration and d_{dist} captures spatial distribution; $d_{\max}$ and $m_{\max}$ are their expected upper bounds, dynamically updated during training to normalise density metrics to the range of $[0, 1]$ across different datasets. For each downsampled point x_i, d_{num} is defined by counting points near p_i that collapse to x_i as:

$$d_{\text{num}}(x_i) = |\{p_j : \text{NN}(p_j) = x_i\}|, \tag{5.5}$$

and d_{dist} measures the total distance between x_i and each original point in the neighbourhood of p_i that collapses to x_i:

$$d_{\text{dist}}(x_i) = \frac{1}{d_{\text{num}}(x_i)} \sum_{p_j \in \text{NN}^{-1}(x_i)} \|p_j - x_i\|_2. \tag{5.6}$$

Here, $\mathrm{NN}(p_j)$ returns the nearest downsampled point to p_j, and $\mathrm{NN}^{-1}(x_i)$ is the set of original points mapped to x_i.

In SemanticKITTI [5] and ShapeNet [6], point clouds exhibit considerable variability in density and distribution, making fixed normalisation parameters inadequate. To address this, this work adopts the Exponential Moving Average (EMA) approach [181], which dynamically updates the normalisation parameters $d_{\max}$ and $m_{\max}$ to reflect the varying density and distance distributions across different datasets. Denote the normalisation parameters for d_{num} and d_{dist} as $\theta \in \{d_{\max}, m_{\max}\}$, and the current training iteration as t. The normalisation parameters $d_{\max}$ and $m_{\max}$ are dynamically updated as:

$$\theta^{(t)} = (1 - \gamma) \cdot \theta^{(t-1)} + \gamma \cdot P_{95}(m^{(t)}), \tag{5.7}$$

to incorporate both the previous parameters $\theta^{(t-1)}$, and the 95th percentile of current batch statistics $P_{95}(m^{(t)})$*, enabling gradual adaptation to dataset characteristics while maintaining stability.

5.3.3.2 Global and Local Variance-based Channel Importance

Point clouds contain complex surface geometries with sharp edges, fine details, and intricate structures, leading to strong spatial variations across feature channels [41, 67]. This work proposes an enhanced channel-importance calculation method that augments variance-based metrics with gradient information while maintaining low computational cost.

Specifically, let the variance feature values in channel c be denoted as Var_c. The gradient metric for the channel, denoted as Grad_c, measures local detail variations

*Experiments across various percentiles (90th-99th) and adaptation rates ($[0.01, 0.5]$) revealed that P_{95} and $\gamma = 0.1$ yield optimal performance.

by summarising the differences between adjacent positions within the channel:

$$\text{Grad}_c = \frac{1}{N-1} \sum_{j=1}^{N-1} |\mathbf{z}_{c,j+1} - \mathbf{z}_{c,j}|, \tag{5.8}$$

where $\mathbf{z}_{c,j}$ is the feature value of channel c at position j in the encoded latent space.

The final channel importance, denoted as $\mathcal{I}_c$, is then computed as a weighted combination of normalised variance and gradient metric in order to capture both globally informative features with high variance and locally discriminative features with high gradient as:

$$\mathcal{I}_c = \beta \cdot \text{norm}(\text{Var}_c) + (1-\beta) \cdot \text{norm}(\text{Grad}_c). \tag{5.9}$$

β is empirically set as 0.6 to balance the relative contribution between global and local components.

5.3.3.3 Density-aware Tail Drop

While ProgDTD [105] extends the tail-drop [170] from a latent feature-only drop strategy to a both latent and hyper-latent [167] drop strategy in 2D image coding, this work proposes tail-drop for both latent features and down-sampled coordinates to achieve better performance, as confirmed by the Ablation Study in Sect. 5.4.5.1.

Learning-based PCC normally relies on two equally critical data modalities: semantic features that encode surface properties and coordinate features that capture spatial geometry [36] [1] [11]. Unlike the hyper-latent in image coding, coordinate information constitutes the core geometric data that directly determines reconstruction quality. Meanwhile, the importance of coordinate features varies across spatial regions, with surfaces typically demonstrating high spatial correlation and redundancy. Thus, this work proposes a density-aware tail drop method that employs a combined drop strategy to achieve this by applying density-guided scalable coding to both features and coordinates based on their channel importance, to ensure geometric consistency between coordinate and semantic representations. Specifically,

given a drop ratio ρ and channel importance of features and coordinates, denoted as $\mathcal{I}_z = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{C_z}\}$ and $\mathcal{I}_{xyz} = \{\mathcal{I}_1^{xyz}, \mathcal{I}_2^{xyz}, \ldots, \mathcal{I}_{C_{xyz}}^{xyz}\}$, respectively, this density-aware tail drop strategy retains the top $(1-\rho)$ fraction of channels in both feature and coordinate spaces, as:

$$\mathbf{z}_\rho = \mathbf{z} \odot \mathbf{M}(\mathcal{I}_z, \rho), \quad \mathbf{z}_{xyz,\rho} = \mathbf{z}_{xyz} \odot \mathbf{M}(\mathcal{I}_{xyz}, \rho). \tag{5.10}$$

Here, $\mathbf{M}(\mathcal{I}, \rho) \in \{0,1\}^C$ is a binary mask that preserves the top $(1-\rho) \times C$ channels according to importance ranking $\mathcal{I}$, C is the total channel count, and $\odot$ denotes element-wise multiplication. This synchronised drop with a common ρ maintains correspondence between feature and coordinate channels throughout the coding process.

5.3.4 Loss Function

Following D-PCC [36], this work adopts an integrated loss $\mathcal{L}$, which is formulated as a weighted sum of geometry reconstruction quality terms and coding efficiency constraints:

$$\mathcal{L} = \mathcal{L}_{\text{CD}} + \sigma \cdot \mathcal{L}_{\text{Dens}} + \omega \cdot \mathcal{L}_{\text{Coord}} + \eta \cdot \mathcal{L}_{\text{Points}} + \lambda \cdot \mathcal{R}_{\text{BPP}}. \tag{5.11}$$

Here, $\mathcal{L}_{\text{CD}}$ measures the Chamfer Distance between the original and reconstructed point clouds, ensuring geometric fidelity, $\mathcal{L}_{\text{Dens}}$ preserves local density distributions, $\mathcal{L}_{\text{Coord}}$ regularises the quantized spatial representation before and after the entropy bottleneck, $\mathcal{L}_{\text{Points}}$ constrains the total number of reconstructed points, and finally, $\mathcal{R}_{\text{BPP}}$ represents the bit rate loss. The weighting coefficients σ, ω, η, and λ control the trade-off between different reconstruction quality aspects and coding efficiency. Consistent with D-PCC [36], we use the same weighting coefficients, enabling fair comparison with baseline methods.

5.3.5 Scalable Coding Evaluation

5.3.5.1 Channel-based Scalable Coding Evaluation

To evaluate the performance of scalable rate distortion, this work conducts channel-based test. Given the importance ranking $\mathcal{I}$ computed from latent features and coordinates, this work simulates different coding ratios by retaining varying ratios of channels. Given a drop ratio ρ, the top $k = \lceil (1-\rho) \cdot C \rceil$ channels are reserved, and the scalable reconstruction becomes:

$$\hat{\mathbf{z}}_\rho = \hat{\mathbf{z}} \odot \mathbf{M}(\mathcal{I}_\ddagger, \rho), \quad \hat{\mathbf{z}}_{xyz,\rho} = \hat{\mathbf{z}}_{xyz} \odot \mathbf{M}(\mathcal{I}_{xyz}, \rho), \tag{5.12}$$

where $\hat{\mathbf{z}}$ and $\hat{\mathbf{z}}_{xyz}$ represent quantized latent features and quantized coordinates after entropy bottleneck compression, respectively (see Eq. 5.2). Empirically, we observe a significant quality improvement when $k \in [1, 13]$, with performance plateauing for $k \geq 16$ in our 32-channel configuration.

5.3.5.2 Bitrate Calculation

Following the entropy-based measurement protocols from D-PCC [36] and H.264 SVC [30], this work computes the effective bits-per-point (BPP) for each scalable level. Given a drop ratio ρ, the BPP is:

$$\mathcal{R}_{\text{BPP}_\rho} = \frac{1}{N} \left(\sum_{i \in \mathcal{S}_\rho} -\log_2 p(\hat{\mathbf{z}}_i) + \sum_{j \in \mathcal{S}_{xyz,\rho}} -\log_2 p(\hat{\mathbf{z}}_{xyz,j}) \right), \tag{5.13}$$

where $\mathcal{S}_\rho$ and $\mathcal{S}_{xyz,\rho}$ denote the sets of retained channel indices for features and coordinates, respectively, and $p(\cdot)$ represents the learned probability distributions from entropy models $\mathcal{B}_{\mathbf{z}}$ and $\mathcal{B}_{\mathbf{z}_{xyz}}$.

5.4 Experiments

5.4.1 Datasets and Implementation

This work adopts two benchmark datasets: SemanticKITTI [5] and ShapeNet-Core.v1 [6]. SemanticKITTI consists of approximately 43,000 LiDAR scan frames

from urban and suburban driving environments, each containing around 120,000 points with higher density on vehicles due to sensor proximity. ShapeNetCore.v1 [6] comprises 51,300 synthetic 3D models across 55 categories, with each model averaging 55,000 points. When preprocessing these datasets for training and testing, this work strictly adheres to the requirements specified in D-PCC [36]. The training objective combines distortion loss, density loss (initialised at 10^{-4}), and coordinate loss (initialised at 5×10^{-5}). Optimisation is performed over 50 epochs using the Adam optimiser, with an initial learning rate of 10^{-3}, decayed by 0.5 every 15 epochs. All experiments are conducted with an NVIDIA A40 GPU.

Based on experimental results, we increased the number of latent feature channels from the original 8 in [36] to 32 to enable effective scalable coding. This provides sufficient channel diversity, enhancing representation capacity and coding efficiency: 16 channels yielded inadequate diversity, while 64 introduced redundancy without commensurate gains, making 32 optimal for quality-efficiency balance. We thus adopted 32 channels across all experiments. Further, unlike D-PCC [36], which optimises at batch size 1, we used 32 for ScalaDAT to reduce computational demands.

5.4.2 Evaluation Metrics

As there is currently no standardization or benchmark for scalable point cloud geometry coding, this work compares its results against other baseline learning-based point cloud coding models on the same datasets, including D-PCC [36], MPEG G-PCC (TMC13 v14.0 under the octree geometry coding configuration) [31], Google Draco [182], MPEG Anchor [183], PCGC [60], Depeco [75], and JPEG Pleno [1].

We evaluate ScalaDAT using a suite of metrics: PSNR-D1, PSNR-D2, Chamfer Distance (CD), and Bjøntegaard Delta Rate (BD-Rate) based on evaluation benchmarks [116, 49], to collectively assess perceptual quality, geometric fidelity, and coding efficiency. As explicitly formulated in Section 2.4 of Chapter 2, these eval-

uation methods assess geometric accuracy, surface smoothness, and shape fidelity, strictly following the JPEG Pleno Common Test Conditions [49] and D-PCC [36] to ensure robust comparability.

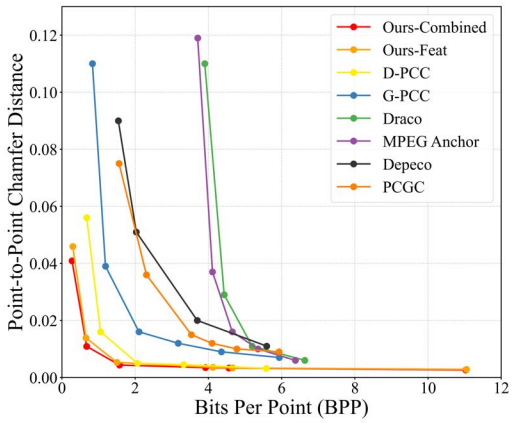
To complement these geometric assessments, BD-Rate is utilized to quantify the overall coding efficiency. It measures the relative average bitrate difference between two coding algorithms across an interpolated rate-distortion curve at matched quality levels (typically using PSNR). A negative BD-Rate indicates a proportional reduction in bitrate for the same visual quality, thereby demonstrating superior compression performance.

5.4.3 Evaluation Results

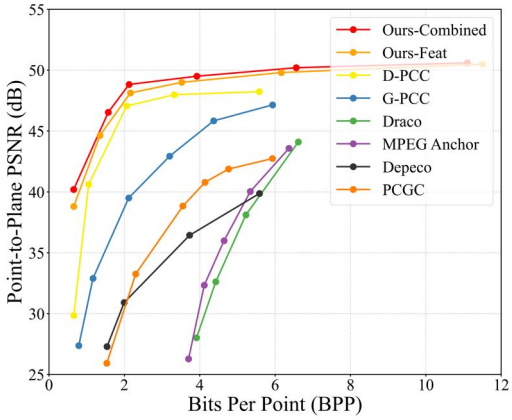
5.4.3.1 Rate Distortion Evaluation

Results demonstrate the superior RD performance on the benchmark datasets SemanticKITTI [5] and ShapeNet [6]. As shown in Fig. 5.3, this approach consistently outperforms baseline methods in quantitative evaluations using CD, PSNR-D1, and PSNR-D2 metrics.

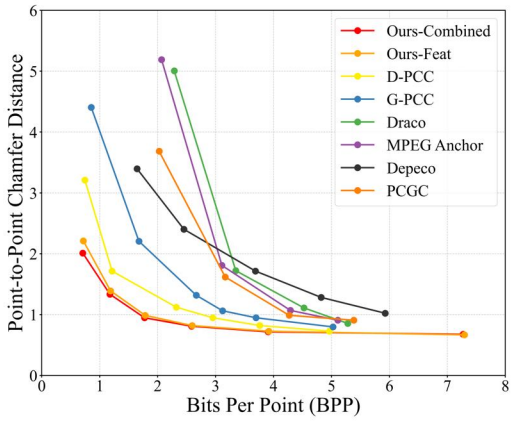
To comprehensively evaluate our method, we report non-scalable results from training and testing without channel drop. Notably, ScalaDAT consistently achieves significantly lower CD values across all bitrates, especially in the low-bitrate range of 0.5–2.0 BPP. For instance, on SemanticKITTI (Fig. 5.3b), it attains $\sim$ 43 dB PSNR-D2 at 1.0 BPP, a quality level that competing methods only reach at substantially higher bitrates. Similarly, on ShapeNet (Fig. 5.3d), it achieves $\sim$ 39.2 dB at 2.0 BPP, underscoring its efficiency in preserving geometric fidelity. Besides quality improvements, ScalaDAT also extends the supported larger bitrate range than D-PCC [36]: from 0–6 BPP to 0–11 BPP on SemanticKITTI and 0–5 BPP to 0–7 BPP on ShapeNet, enhancing flexibility for bandwidth-limited applications. Moreover, the BD-rate metric shows that our combined Drop method achieves 28.6% bitrate



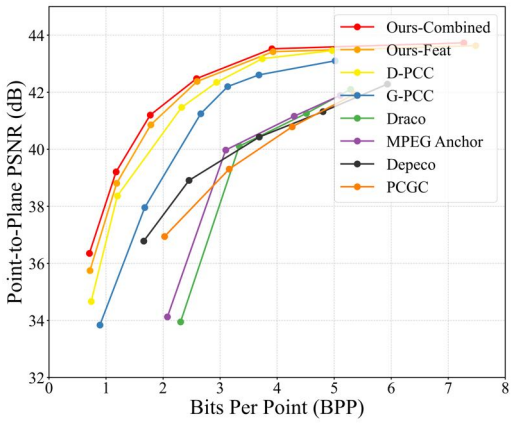
(a) SemanticKITTI: BPP vs Chamfer Distance



(b) SemanticKITTI: BPP vs PSNR



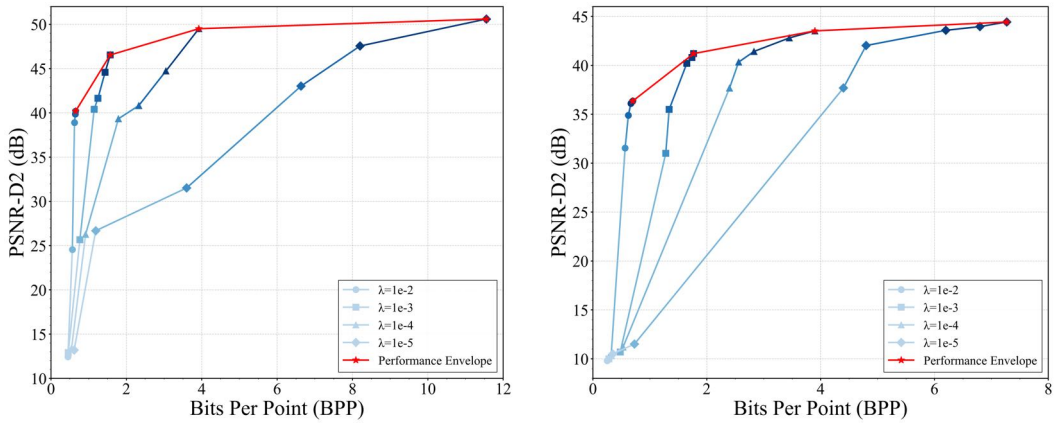
(c) ShapeNet: BPP vs Chamfer Distance



(d) ShapeNet: BPP vs PSNR

Figure 5.3 : Quantitative results of non-scalable ScalaDAT on SemanticKITTI and ShapeNet: BPP vs. CD (a, c) and BPP vs. PSNR-D2 (b, d). Models are trained using scalable coding. Results compare two tail-drop strategies: Combined Drop (red lines) and Feature-Only Drop (orange lines), with the model trained using scalable coding. The differences between these strategies will be described in terms of the Ablation Study below.

savings on SemanticKITTI and 18.15% on ShapeNet relative to D-PCC [36], at equivalent quality levels. This consistent performance across structured outdoor environments and object-centric data highlights the versatility and robustness of the proposed method for scalable point cloud coding.



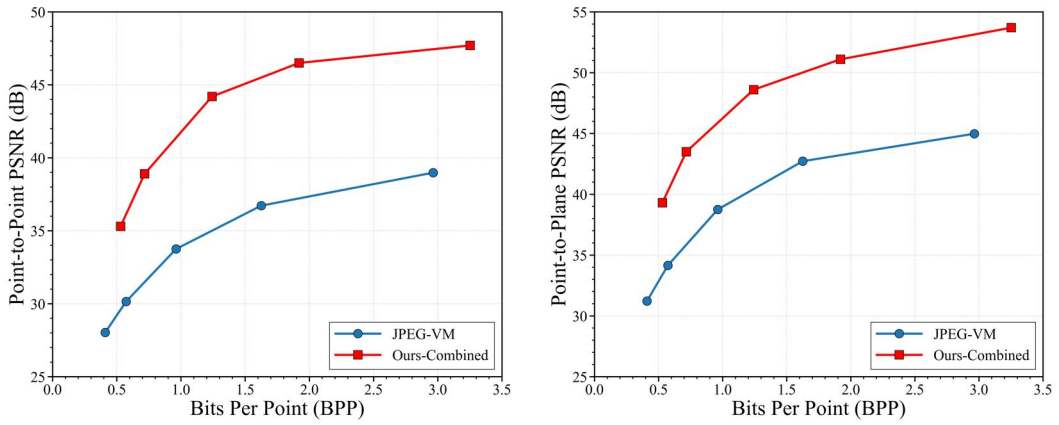
(a) SemanticKITTI (b) ShapeNet

Figure 5.4 : Scalable performance of ScalaDAT (PSNR-D2 vs BPP) on SemanticKITTI and ShapeNet across different λ values with simultaneous latent feature and coordinate drop.

To benchmark the ScalaDAT model against the JPEG-VM standard [1] for learning-based PCC, this work utilized the SemanticKITTI dataset [5]. Point clouds were voxelised at a 10-bit precision to balance geometric fidelity and coding efficiency. Each LiDAR point cloud was normalised to a unit cube via scene-adaptive bounding boxes, with outliers removed and 5% padding to reduce boundary artifacts, and then quantized to integers in [0, 1023]. The preprocessed dataset was evaluated using checkpoints from both ScalaDAT and JPEG-VM [1]. As shown in Fig. 5.5, ScalaDAT achieves substantial gains over JPEG-VM, with BD-rate reductions exceeding 65% for PSNR-D1 and 55% for PSNR-D2 on SemanticKITTI. These BD-rate savings are largely attributable to evaluating JPEG-VM with publicly released pre-trained models on SemanticKITTI without retraining.

5.4.3.2 Scalable Performance on PSNR-D2

To evaluate the performance of this work, we utilise PSNR-D2, and the result is shown in Fig. 5.5.



(a) BPP vs PSNR-D1 (b) BPP vs PSNR-D2

Figure 5.5 : Quantitative comparison of non-scalable ScalaDAT with JPEG-VM [1] on SemanticKITTI in terms of PSNR-D1 and PSNR-D2 for each point cloud model.

This work evaluates scalable coding using the **PSNR-D2** metric, following the de facto benchmark for point-cloud compression [36]. Figs. 5.4a and 5.4b show the rate–distortion (RD) trade-off for SemanticKITTI and ShapeNet, respectively. Each plot includes multiple curves for different values of the trade-off parameter λ that controls the balance between coding efficiency and reconstruction quality.

In detail, on SemanticKITTI (Fig. 5.4a), PSNR-D2 starts relatively high even at low bits (0–1 BPP) and increases gradually; Once BPP exceeds 1, it rises more steeply and tends to saturate around 4–6 BPP. PSNR-D2 starts relatively high even at low bit budgets (0–1 BPP) and increases gradually; once BPP exceeds ~ 1, it rises more steeply and tends to saturate around 4–6 BPP. Conversely, on ShapeNet (Fig. 5.4b), PSNR-D2 begins lower at small BPP and improves more slowly. In both datasets, we observe a critical threshold near 1 BPP beyond which

additional bits yield markedly better reconstruction quality, consistent with higher scalable gains. These differences reflect intrinsic point-cloud characteristics: SemanticKITTI's LiDAR-derived regular structure and lower complexity enable efficient coding and superior quality at low bitrates (*e.g.*, 0–1 BPP), whereas ShapeNet's diverse, intricate objects require more bits to achieve comparable quality, particularly evident in the low-BPP regime, where SemanticKITTI achieves robust reconstruction with fewer bits.

5.4.3.3 Computational Complexity Analysis

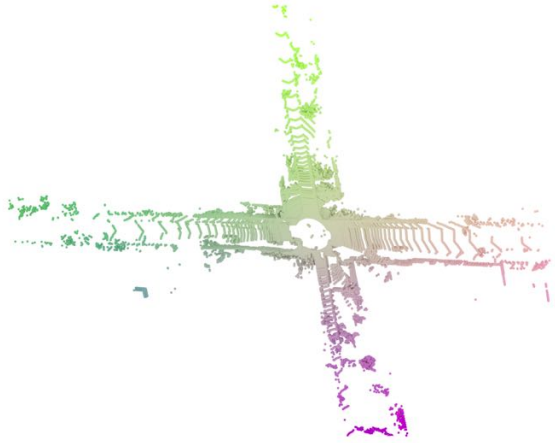
To address the operational feasibility of the proposed SPCC framework, a computational complexity analysis was conducted compared to the single-rate baseline D-PCC [36]. A critical advantage of ScalaDAT lies in its training efficiency; unlike prior scalable methods that require complex multi-stage recursive training for different quality layers, ScalaDAT achieves full scalable capabilities through a single training iteration.

Regarding inference complexity tested on an NVIDIA A40 GPU, to support fine-grained scalable channel dropping, ScalaDAT utilises an expanded 32-channel configuration, which naturally results in a moderately larger model size ($\sim$45 MB) compared to the 8-channel baseline. While the full-channel reconstruction (SR = 1.0) requires comparable encoding and decoding latency to the baseline, operating at lower Scalable Ratios dynamically reduces the computational load on the decoder side. Specifically, as the SR decreases, the density-aware tail-drop mechanism prunes unselected latent channels before the decoder phase. Consequently, the decoder physically bypasses the inverse convolution operations for the dropped channels, leading to a measurable reduction in decoding latency compared to processing a full monolithic bitstream. This dynamically scaling computational load perfectly aligns with the requirements of edge devices in bandwidth-constrained and power-limited

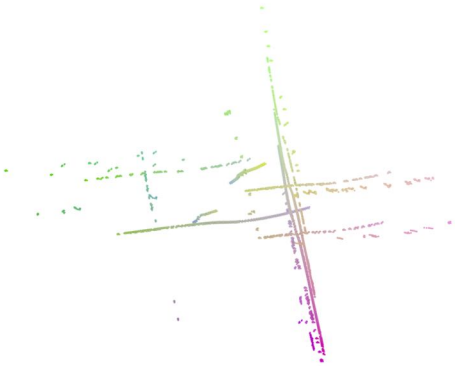
environments.

5.4.4 Visualisation

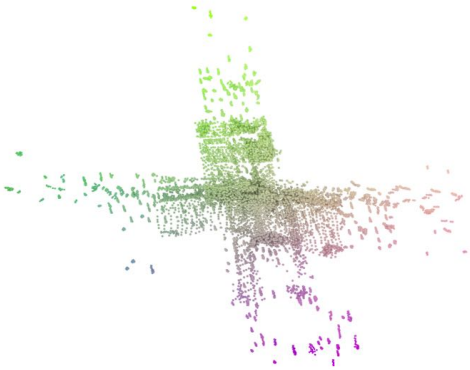
Figs. 5.6 and 5.7 depict scalable reconstruction of point clouds from SemanticKITTI and ShapeNet datasets when $\lambda = 0.001$. For instance, in Fig. 5.6, at a Scalable Ratio (SR) of 0.03, the essential scene structure remains discernible, preserving core geometry with a minimal bitstream. As SR increases, the decoder incrementally activates additional latent channels according to our density-aware importance ranking (from highest to lowest geometric significance). This prioritization yields rapid gains in reconstruction quality, highlighting the effectiveness of the proposed density-aware channel importance assessment in favouring geometrically critical coordinates and features for scalable transmission. Notably, Fig. 5.7 features two objects selected for their sparse-to-dense point distributions. At very low SR, reconstructions may exhibit minor spatial overlap and noise; we hypothesise that as SR increases and additional components of the bitstream are decoded, the enriched latent features improve point localisation while suppressing spurious and overlapping points, thereby improving overall reconstruction performance.



Ground Truth

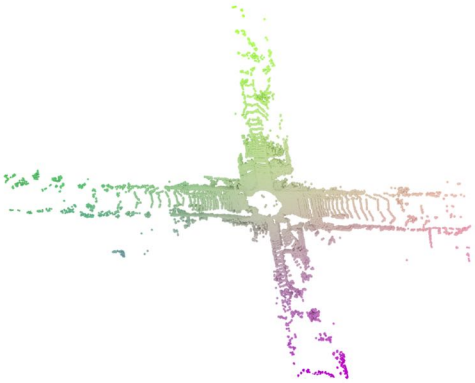


SR=0.03



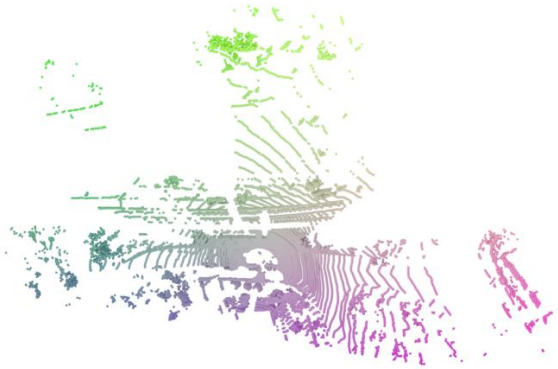
SR=0.09

SR=0.3

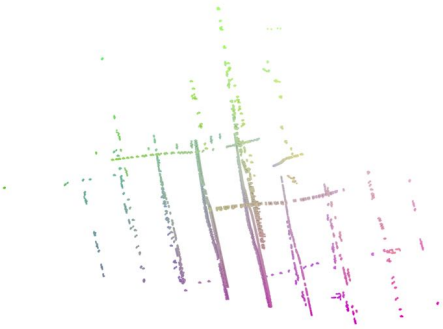


SR=0.5

Figure 5.6 : Visualisation of the scalable coding applied to two SemanticKITTI models [5] with varying SR values.

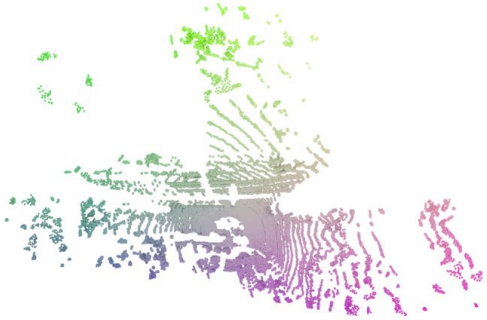


Ground Truth



SR=0.03

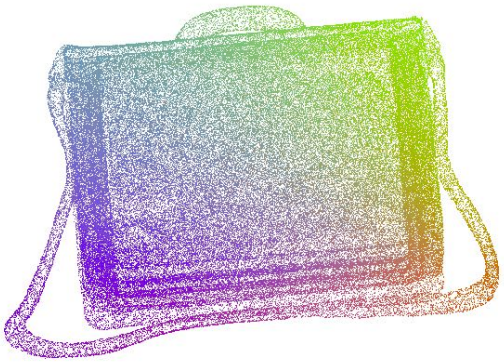
SR=0.09



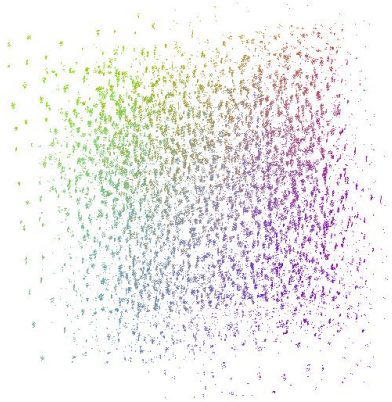
SR=0.3

SR=0.5

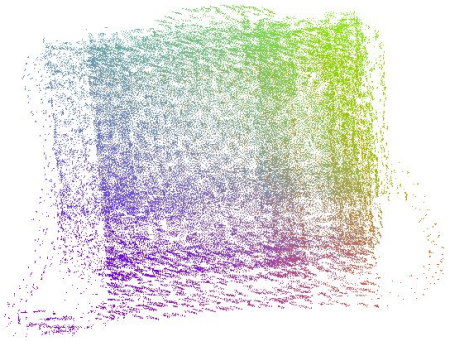
Figure 5.6 : Visualization of the scalable coding (continued).



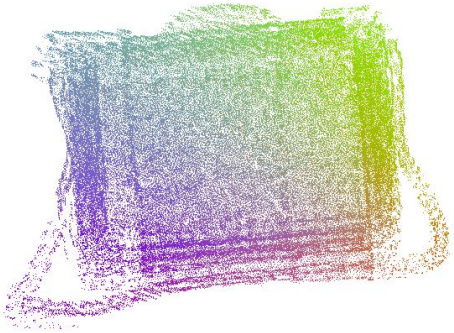
Ground Truth



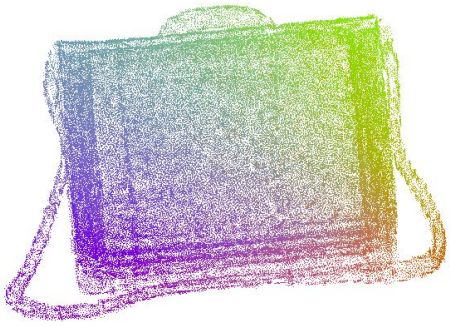
SR=0.09



SR=0.12



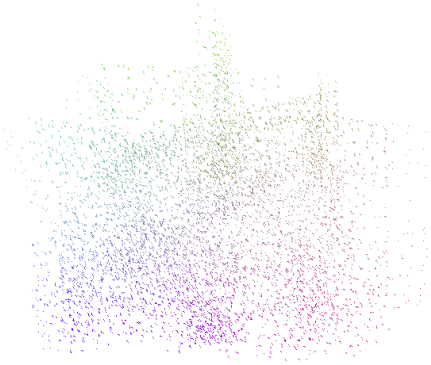
SR=0.19



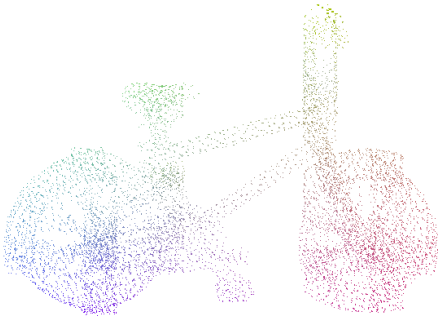
SR=0.25

Figure 5.7 : Visualisation of the scalable coding applied to two ShapeNet models [6] with lambda as 0.001, illustrating the impact of combined feature and coordinate dropping. From left to right in each 2x2 grid: Scalable Ratio (SR)=0.09, 0.12, 0.19, and 0.25, corresponding to drop rates (1-PR) of 0.91, 0.88, 0.81, and 0.75, respectively.

Ground Truth



SR=0.09



SR=0.12

SR=0.19

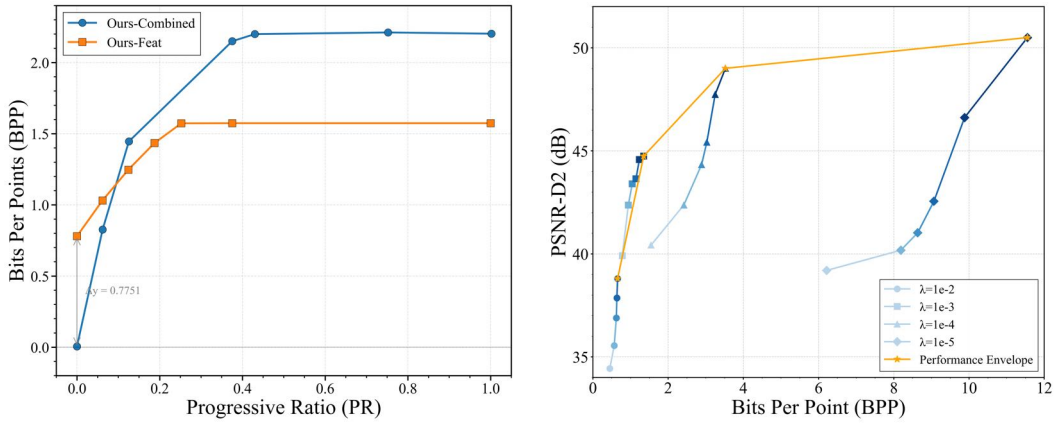
SR=0.25

Figure 5.7 : Visualization of the scalable coding (continued).

5.4.5 Ablation Studies

5.4.5.1 Ablation Study on Different Dropping Strategies

This work compares two strategies: combined coordinate-feature drop and feature-only drop, where models were trained with $\lambda \in [10^{-2}, 10^{-5}]$. Each model was trained once per λ and subsequently evaluated through scalable testing across multiple scalable ratios. Results are displayed in Fig. 5.3, Fig. 5.4a, and Fig. 5.8b, which indicate the higher PSNR-D2 performance for the combined-drop method, underscoring its effectiveness through BD-Rate improvements exceeding 12.3% on the SemanticKITTI dataset and 9.6% on the ShapeNet dataset compared to the feature-only strategy statistically.

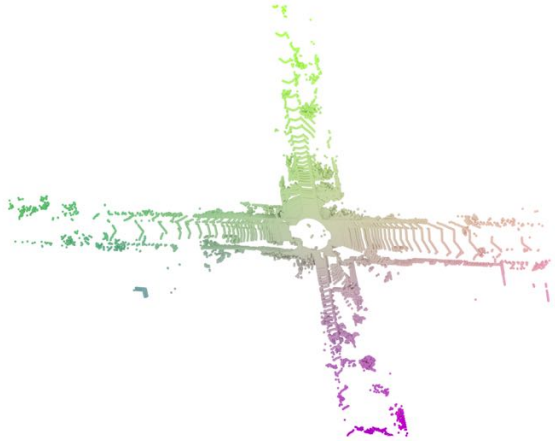


(a) (b)

Figure 5.8 : Ablation study of scalable coding for different drop strategies and λ values on SemanticKITTI: (a) BPP vs PR for different drop strategies, and (b) PSNR-D2 vs BPP for feature-only drop with varying λ values.

Besides, we set $\lambda = 0.001$ here to show the visualisations of different drop strategies' results and get Fig. 5.9, which is consistent with the setting of Sect. 5.4.4. Compared with Fig. 5.6, even though the feature-only drop shows strong central and structural information when SR is low (*e.g.*, 0.03), its BPP starts from a much higher value, which further proves the benefit of the combined drop that it can

achieve the scalable coding starting from a much smaller BPP value. In detail, the feature-only method shows an elevated initial BPP of 0.77 because of direct coordinate encoding, in contrast to the combined-drop approach starting at 0.008 BPP, which can be observed from the BPP-SR as shown in Fig. 5.8a.



Ground Truth

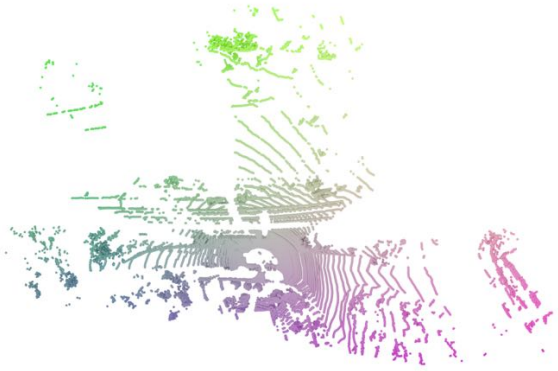
SR=0.03

SR=0.09

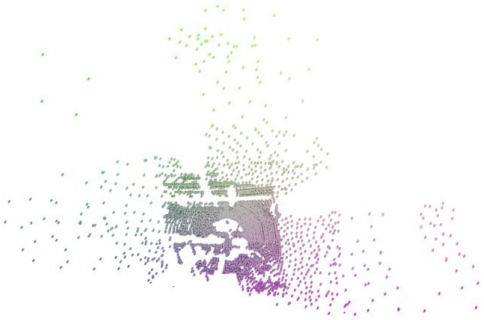
SR=0.18

SR=0.36

Figure 5.9 : Visualisation of scalable coding with feature-only dropping applied to two SemanticKITTI models. From left to right in each 2x2 grid: SR=0.03, 0.09, 0.18, and 0.36, corresponding to drop rates (1-PR) of 0.97, 0.91, 0.82, and 0.64, respectively.

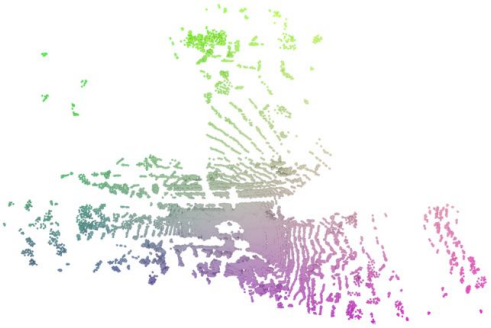


Ground Truth

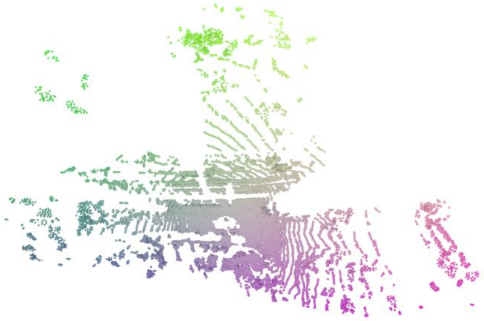


SR=0.03

SR=0.09



SR=0.18



SR=0.36

Figure 5.9 : Visualization of scalable coding with feature-only dropping (continued).

5.5 Summary

This work presented ScalaDAT, a new framework for scalable point cloud coding that introduces Tail-drop, an new data selection method that strategically discards less critical information while maintaining reconstruction quality. Unlike conventional methods, ScalaDAT achieves scalable coding with just a single training iteration, significantly reducing computational overhead while maintaining competitive performance. Comprehensive experiments conducted on SemanticKITTI and ShapeNet datasets demonstrate superior BD-Rate performance compared to baseline methods. Our exploration of density-aware Tail-drop and latent space drop strategies provides new insights into coding dynamics, revealing opportunities for further optimisation.

Chapter 6

Conclusions and Future Work

This thesis has addressed critical challenges in point cloud compression (PCC), motivated by the growing demands of applications such as extended reality (XR), autonomous driving, and scene reconstruction, where point clouds' unstructured and voluminous nature necessitates efficient techniques to balance storage, transmission, and fidelity while supporting downstream tasks like classification and scalable decoding. Through a comprehensive exploration of learning-based methods, it has developed new frameworks that develop robust frameworks aimed at improving in rate-distortion (RD) performance, low-bitrate accuracy, and scalability. Specifically, integrating proposed contributions from Chapter 3 to Chapter 5—such as Transformer-based RD gains, edge-preserving classification, and density-aware scalable coding—to synthesise how these advancements resolve the foundational challenges and gaps outlined in Chapters 1 and 2. It revisits key elements like representations (*e.g.*, point-based and voxel-based), datasets (*e.g.*, ShapeNet, ModelNet40), and metrics (*e.g.*, BD-Rate, BD-Top-1 Accuracy) to evaluate overall impact, while proposing adaptive future directions to build on the logical progression toward versatile, high-fidelity PCC.

6.1 Summary of Key Findings and Contributions

The research began with an in-depth background on the importance of point clouds and the motivations for compression, highlighting gaps in traditional methods like MPEG G-PCC [31] and V-PCC [32], which struggle with irregular densities and real world demands. We defined the PCC problem, emphasising the rate-distortion

trade-off, and outlined objectives focused on transformer-based enhancements, edge-preserving classification in compressed domains, and density-aware scalable coding. In Chapter 3, we introduced a transformer-based geometric PCC method with local neighbour aggregation. By integrating global attention for long-range dependencies and local modules to mitigate down-sampling losses, this approach preserved structural details and managed decoding times. The method demonstrated measurable BD-Rate improvements on ShapeNetCore.v2 compared to baseline point-based methods like [38, 56], offering an effective strategy for geometric reconstruction.

Chapter 4 presented CPCC-PES, a compressed point cloud classification framework with attention-based edge sampling. This method improved BD-Rate and Top-1 accuracy on ModelNet40 [6] relative to existing approaches like LPCCC [2] at low bitrates. By enriching bitstreams with boundary semantics, the edge-centric strategy provided a robust alternative to traditional uniform sampling techniques such as farthest point sampling (FPS), enhancing feature retention for downstream tasks.

In Chapter 5, we proposed ScalaDAT, a density-aware tail-drop framework for scalable PCC, designed to enable single-iteration training with incremental decoding. Validated on SemanticKITTI [5] and ShapeNet [6], ScalaDAT maintained competitive RD performance by prioritising high-density regions and synchronising feature-coordinate dropping. This provides a practical and scalable coding solution tailored for bandwidth-constrained scenarios.

Collectively, these contributions provide structured enhancements to learned PCC: measurable RD improvements in geometric coding, increased low-bitrate classification accuracy, and an efficient scalable framework across diverse datasets. The literature review in Chapter 2 contextualised these developments, revealing cross-domain influences from image and video compression and identifying persistent

structural gaps in point-based, octree-based, and voxel-based paradigms.

6.2 Implications and Significance

The developed methods have broad implications for real-world applications. By improving RD performance and enabling scalable decoding, this work facilitates efficient data handling in resource-limited environments, such as mobile AR/VR and autonomous systems. The density-aware and edge-preserving techniques enhance fidelity in sparse, irregular data, bridging gaps between compression and analysis tasks. Furthermore, the single-training scalable paradigm reduces computational overhead, promoting sustainability in large-scale 3D processing. These advancements contribute to the evolving field of PCC, aligning with emerging standards like JPEG Pleno [1] and supporting interdisciplinary applications in robotics, cultural heritage, and remote sensing. While the thesis focused on geometry compression, extensions to attributes like colour and intensity were explored, showing potential for joint coding. Limitations include dataset-specific optimisations and computational demands for ultra-large clouds, which could be mitigated through hardware acceleration.

6.3 Future Directions

6.3.1 Limitations of Current Contributions

While the proposed methods demonstrate competitive performance on their respective benchmarks, several limitations should be acknowledged:

- **Chapter 3 (Transformer-based PCC):** The method is validated exclusively on ShapeNetCore.v2, a dense synthetic object dataset. Its generalisation to sparse LiDAR scans or dynamic sequences remains unverified. Furthermore, the Local Neighbour Aggregation module increases decoding latency by ap-

proximately 2.5× compared to the baseline [38], which may limit applicability in latency-sensitive deployment scenarios. An iso-latency comparison was not feasible within the current architectural design, as the LNA module cannot be partially disabled without retraining.

- **Chapter 4 (Edge-based CPCC):** The evaluation on ModelNet40 focuses on isolated object-level classification. Extending to scene-level tasks (*e.g.*, semantic segmentation on KITTI) would better demonstrate practical utility. Additionally, the comparison with DPCC baselines involves an inherent asymmetry: the proposed CPCC-PES is trained end-to-end for classification, whereas generic codecs like G-PCC and OctAttention are not retrained on compression artifacts, which may overstate the relative advantage.

- **Chapter 5 (ScalaDAT):** The single-training scalable paradigm, while computationally efficient, constrains the model to a fixed channel-importance ranking determined at training time. This ranking may not generalise optimally across all point cloud categories or density distributions unseen during training.

6.3.2 Impact of Higher-Quality Point Clouds

As 3D acquisition technologies advance, point clouds with substantially higher point counts, finer spatial resolution, and richer attribute information are becoming increasingly available. The impact on each proposed method would be as follows:

- **Transformer-based PCC (Ch. 3):** Higher-density inputs would increase the computational cost of both FPS down-sampling and KNN-based neighbour search in the LNA module, scaling approximately quadratically with point count. However, the richer local neighbourhoods could also improve the quality of density and position embeddings, potentially yielding greater BD-

Rate gains. Adapting the architecture to handle variable input sizes efficiently (*e.g.*, through hierarchical patch-based processing) would be necessary.

- **Edge-based Classification (Ch. 4):** Denser point clouds provide more precise edge and boundary information, which could directly benefit the attention-based edge sampling module by offering finer-grained structural features for classification. The three-stage down-sampling pipeline may need additional stages or adaptive ratios to handle the increased input scale without losing critical edge details

- **ScalaDAT (Ch. 5):** Higher-quality inputs with greater density variation would amplify the advantage of density-aware tail-drop, as the contrast between structurally significant high-density regions and sparse background becomes more pronounced. However, the fixed 32-channel latent space may become a bottleneck for representing the increased geometric complexity, suggesting that adaptive channel allocation or a larger latent dimensionality could be explored.

6.3.3 Other Future Directions

Beyond addressing the above limitations, several broader research directions emerge from this work. Extending ScalaDAT to joint geometry-attribute scalable coding and incorporating temporal redundancies for dynamic point cloud sequences represent natural next steps. Integrating multi-modal data (*e.g.*, semantic labels from LiDAR) and exploring federated learning for privacy-preserving compression are promising avenues. Additionally, real-world deployment on edge devices and standardisation efforts for scalable coding benchmarks would advance practical adoption. Investigating hybrid Transformer-CNN architectures and adaptive entropy models could further optimise the latency-quality trade-off across all three proposed frameworks.

Bibliography

[1] A. F. Guarda, N. M. Rodrigues, M. Ruivo, L. Coelho, A. Seleem, and F. Pereira, "IT/IST/IPLeiria response to the call for proposals on JPEG pleno point cloud coding," *arXiv preprint arXiv:2208.02716*, 2022.

[2] M. Ulhaq and I. V. Bajić, "Learned point cloud compression for classification," *IEEE 25th International Workshop on Multimedia Signal Processing*, 2023.

[3] M. Quach, J. Pang, D. Tian, G. Valenzise, and F. Dufaux, "Survey on deep learning-based point cloud compression," *Frontiers in Signal Processing*, vol. 2, p. 846972, 2022.

[4] B. Yang, N. Haala, and Z. Dong, "Progress and perspectives of point cloud intelligence," *Geo-spatial Information Science*, vol. 26, no. 2, pp. 189–205, 2023.

[5] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9297–9307.

[6] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[7] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning

for 3D point clouds: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, 2020.

[8] F. Poux, R. Neuville, P. Hallot, and R. Billen, "Smart point cloud: definition and remaining challenges," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, no. W1, 2016.

[9] M. Wang, R. Huang, W. Xie, Z. Ma, and S. Ma, "Compression approaches for LiDAR point clouds and beyond: A survey," *ACM Transactions on Multimedia Computing, Communications and Applications*, 2025.

[10] Y. Xu, Y. Zhang, S. Xia, K. Yang, H. Huang, Z. Shan, W. Huang, Q. Yang, and L. Yang, "Point cloud compression and objective quality assessment: A survey," *arXiv preprint arXiv:2506.22902*, 2025.

[11] Z. Luo, W. Jia, and S. Perry, "Transformer-based geometric point cloud compression with local neighbor aggregation," in *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2023, pp. 223–228.

[12] ——, "Compressed point cloud classification with point-based edge sampling," *EURASIP Journal on Image and Video Processing*, vol. 2024, no. 1, p. 18, 2024.

[13] H. Liu, H. Yuan, Q. Liu, J. Hou, and J. Liu, "A comprehensive study and comparison of core technologies for MPEG 3-D point cloud compression," *IEEE Transactions on Broadcasting*, vol. 66, no. 3, pp. 701–717, 2019.

[14] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli, "A LiDAR point cloud generator: from a virtual world to autonomous driving," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, 2018, pp. 458–464.

[15] T. Fukuda, Y. Zhu, and N. Yabuki, "Point cloud stream on spatial mixed reality-toward telepresence in architectural field," in *Proceedings of the 36th International Conference on Education and Research in Computer Aided Architectural Design in Europe (eCAADe)[Volume 2]*. eCAADe, 2018.

[16] C. Loop, Q. Cai, S. O. Escolano, and P. A. Chou, "JPEG pleno database: Microsoft voxelized upper bodies-a voxelized point cloud dataset," *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012*, 2021.

[17] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "JPEG pleno database: 8i voxelized full bodies (8iVFB v2)-a dynamic voxelized point cloud dataset," 2019.

[18] Alterpex, "What are point clouds?" https://alterpex.com/blog/what-are-point-clouds, 2024, accessed: 2026-02-22.

[19] Ouster, Inc., "Gemini: Spatial intelligence platform," https://ouster.com/products/software/gemini, 2026, accessed: 2026-02-22.

[20] Z. Yang, L. Chen, Y. Sun, and H. Li, "Visual point cloud forecasting enables scalable autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 673–14 684.

[21] H. Arroyo, P. Keir, D. Angus, S. Matalonga, S. Georgiev, M. Goli, G. Dooly, and J. Riordan, "Segmentation of drone collision hazards in airborne radar point clouds using PointNet," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[22] S. Chen, Z. Fang, S. Wan, T. Zhou, C. Chen, M. Wang, and Q. Li, "Geometrically aware transformer for point cloud analysis," *Scientific Reports*, vol. 15, no. 1, p. 16545, 2025.

[23] T. Hackel, N. Savinov, L. Ladicky, J. Wegner, K. Schindler, and M. Pollefeys, "SEMANTIC3D. NeT: A new large-scale point cloud classification benchmark," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, pp. 91–98, 2017.

[24] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[25] J. Wang, B. Fei, D. d. S. Edirimuni, Z. Liu, Y. He, and X. Lu, "A survey of deep learning-based point cloud denoising," *arXiv preprint arXiv:2508.17011*, 2025.

[26] M. Quach, G. Valenzise, and F. Dufaux, "Learning convolutional transforms for lossy point cloud geometry compression," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 4320–4324.

[27] Q. Hu, B. Yang, S. Khalid, W. Xiao, N. Trigoni, and A. Markham, "Sensaturban: Learning semantics from urban-scale photogrammetric point clouds," *International Journal of Computer Vision*, vol. 130, no. 2, pp. 316–343, 2022.

[28] R. Xue, J. Wang, and Z. Ma, "Efficient LiDAR point cloud geometry compression through neighborhood point attention," *arXiv preprint arXiv:2208.12573*, 2022.

[29] Z. Que, G. Lu, and D. Xu, "VoxelContext-Net: An octree based framework for point cloud compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6042–6051.

[30] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Transactions on Circuits and*

Systems for Video Technology, vol. 13, no. 7, pp. 560–576, 2003.

[31] W. MDGC, "7, "G-PCC codec description v9"," *MPEG-3DG. G-PCC Codec Description v9. ISO/IEC JTC1/SC29/WG7 N0011*, 2020.

[32] ——, "7, "V-PCC codec description v12"," *MPEG-3DG. V-PCC Codec Description v12. ISO/IEC JTC1/SC29/WG7 N0012*, 2020.

[33] Y. Cao, Y. Wang, and H. Chen, "Real-time LiDAR point cloud compression and transmission for resource-constrained robots," *arXiv preprint arXiv:2502.06123*, 2025.

[34] N. A. Martins, L. A. d. S. Cruz, and F. Lopes, "Impact of LiDAR point cloud compression on 3D object detection evaluated on the KITTI dataset," *EURASIP Journal on Image and Video Processing*, vol. 2024, no. 1, p. 15, 2024.

[35] J. Dybedal, A. Aalerud, and G. Hovland, "Embedded processing and compression of 3D sensor data for large scale industrial environments," *Sensors*, vol. 19, no. 3, p. 636, 2019.

[36] Y. He, X. Ren, D. Tang, Y. Zhang, X. Xue, and Y. Fu, "Density-preserving deep point cloud compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2333–2342.

[37] J. Wang, D. Ding, Z. Li, X. Feng, C. Cao, and Z. Ma, "Sparse tensor-based multiscale representation for point cloud geometry compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[38] J. Zhang, G. Liu, D. Ding, and Z. Ma, "Transformer and upsampling-based point cloud compression," in *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis*, 2022, pp. 33–39.

[39] C. Fu, G. Li, R. Song, W. Gao, and S. Liu, "Octattention: Octree-based large-scale contexts model for point cloud compression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 625–633.

[40] L. Huang, S. Wang, K. Wong, J. Liu, and R. Urtasun, "Octsqueeze: Octree-structured entropy model for LiDAR compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1313–1323.

[41] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[42] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[43] M. Cui, J. Long, M. Feng, B. Li, and H. Kai, "Octformer: Efficient octree-based transformer for point cloud compression with local enhancement," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, 2023, pp. 470–478.

[44] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. 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*, vol. 9, p. e13, 2020.

[45] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li *et al.*, "Emerging MPEG standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2018.

[46] D. T. Nguyen, M. Quach, G. Valenzise, and P. Duhamel, "Learning-based lossless compression of 3D point cloud geometry," in *ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4220–4224.

[47] ——, "Lossless coding of point cloud geometry using a deep generative model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4617–4629, 2021.

[48] C. Cao, M. Preda, V. Zakharchenko, E. S. Jang, and T. Zaharia, "Compression of sparse and dense dynamic point clouds—methods and standards," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1537–1558, 2021.

[49] S. Perry, "JPEG pleno point cloud coding common test conditions v3.2," *ISO/IEC JTC1/SC29/WG1 N*, vol. 86044, 2020.

[50] S. Perry, H. P. Cong, L. A. da Silva Cruz, J. Prazeres, M. Pereira, A. Pinheiro, E. Dumic, E. Alexiou, and T. Ebrahimi, "Quality evaluation of static point clouds encoded using MPEG codecs," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3428–3432.

[51] A. F. Guarda, N. M. Rodrigues, and F. Pereira, "Point cloud coding: Adopting a deep learning-based approach," in *2019 Picture Coding Symposium (PCS)*. IEEE, 2019, pp. 1–5.

[52] J. Wang, D. Ding, Z. Li, and Z. Ma, "Multiscale point cloud geometry compression," in *2021 Data Compression Conference (DCC)*. IEEE, 2021, pp. 73–82.

[53] J. Wang, R. Xue, J. Li, D. Ding, Y. Lin, and Z. Ma, "A versatile point cloud compressor using universal multiscale conditional coding–part i: Geometry,"

IEEE transactions on pattern analysis and machine intelligence, vol. 47, no. 1, pp. 269–287, 2024.

[54] M. Rudolph, A. Riemenschneider, and A. Rizk, "Progressive coding for deep learning based point cloud attribute compression," in *Proceedings of the 16th International Workshop on Immersive Mixed and Virtual Environment Systems*, 2024, pp. 78–84.

[55] T. Huang and Y. Liu, "3D point cloud geometry compression on deep learning," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 890–898.

[56] W. Yan, S. Liu, T. H. Li, Z. Li, G. Li *et al.*, "Deep autoencoder-based lossy geometry compression for point clouds," *arXiv preprint arXiv:1905.03691*, 2019.

[57] I. Lang, A. Manor, and S. Avidan, "SampleNet: Differentiable point cloud sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7578–7588.

[58] X. Wang, Y. Jin, Y. Cen, T. Wang, B. Tang, and Y. Li, "LightN: Lightweight transformer network for performance-overhead tradeoff in point cloud downsampling," *IEEE Transactions on Multimedia*, 2023.

[59] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[60] J. Wang, H. Zhu, H. Liu, and Z. Ma, "Lossy point cloud geometry compression via end-to-end learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4909–4923, 2021.

[61] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1305–1315, 1997.

[62] Y. Huang, J. Peng, C.-C. J. Kuo, and M. Gopi, "Octree-based progressive geometry coding of point clouds." in *PBG@ SIGGRAPH*, 2006, pp. 103–110.

[63] Y. Jin, Z. Zhu, T. Xu, Y. Lin, and Y. Wang, "ECM-OPCC: Efficient context model for octree-based point cloud compression," in *ICASSP 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 7985–7989.

[64] C. Choy, J. Gwak, and S. Savarese, "4D Spatio-temporal ConvNets: Minkowski convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

[65] C. Cao, "3d point cloud compression," Ph.D. dissertation, Institut Polytechnique de Paris, 2021.

[66] Z. Wang, S. Wan, and L. Wei, "Optimized octree codec for geometry-based point cloud compression," *Signal, Image and Video Processing*, vol. 18, no. 1, pp. 761–772, 2024.

[67] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, pp. 1–12, 2019.

[68] X. Zhang and W. Gao, "Adaptive geometry partition for point cloud compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4561–4574, 2021.

[69] B. Vishwanath, K. Zhang, and L. Zhang, "Advances in predictive RAHT for geometric point cloud compression," *IEEE Transactions on Image Processing*, 2025.

[70] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[71] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Computational Visual Media*, vol. 7, pp. 187–199, 2021.

[72] Z. Liang and F. Liang, "TransPCC: Towards deep point cloud compression via transformers," in *Proceedings of the 2022 International Conference on Multimedia Retrieval*, 2022, pp. 1–5.

[73] C. Chen, Y. Wu, Q. Dai, H.-Y. Zhou, M. Xu, S. Yang, X. Han, and Y. Yu, "A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[74] X. Wen, X. Wang, J. Hou, L. Ma, Y. Zhou, and J. Jiang, "Lossy geometry compression of 3D point cloud data via an adaptive octree-guided network," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.

[75] L. Wiesmann, A. Milioto, X. Chen, C. Stachniss, and J. Behley, "Deep compression for dense point cloud maps," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2060–2067, 2021.

[76] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6411–6420.

[77] L. Gao, T. Fan, J. Wan, Y. Xu, J. Sun, and Z. Ma, "Point cloud geometry compression via neural graph sampling," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 3373–3377.

[78] L. Xie, W. Gao, H. Zheng, and G. Li, "Roi-guided point cloud geometry compression towards human and machine vision," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 3741–3750.

[79] R. Cui, X. Song, W. Sun, S. Wang, W. Liu, S. Chen, T. Shang, Y. Li, N. Barnes, H. Li *et al.*, "LAM3D: Large image-point clouds alignment model for 3D reconstruction from single image," *Advances in Neural Information Processing Systems*, vol. 37, pp. 4454–4480, 2024.

[80] H. Xu, X. Zhang, and X. Wu, "Fast point cloud geometry compression with context-based residual coding and inr-based refinement," in *European Conference on Computer Vision*. Springer, 2025, pp. 270–288.

[81] J. Zhang, T. Chen, D. Ding, and Z. Ma, "G-PCC++: Enhanced geometry-based point cloud compression," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 1352–1363.

[82] ——, "YOGA: Yet another geometry-based point cloud compressor," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 9070–9081.

[83] T. Fan, L. Gao, Y. Xu, D. Wang, and Z. Li, "Multiscale latent-guided entropy model for LiDAR point cloud compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 12, pp. 7857–7869, 2023.

[84] R. Huang, G. Wang, and W. Zhang, "Density-adaptive octree-based point cloud geometry compression," in *2024 International Conference on Ubiquitous Communication (Ucom)*. IEEE, 2024, pp. 227–231.

[85] R. Song, C. Fu, S. Liu, and G. Li, "Efficient hierarchical entropy model for learned point cloud compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 368–14 377.

[86] X. Wang, Y. Zhang, T. Liu, X. Liu, K. Xu, J. Wan, Y. Guo, and H. Wang, "TopNet: Transformer-efficient occupancy prediction network for octree-structured point cloud geometry compression," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 27 305–27 314.

[87] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[88] D. T. Nguyen, M. Quach, G. Valenzise, and P. Duhamel, "Multiscale deep context modeling for lossless point cloud geometry compression," in *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2021, pp. 1–6.

[89] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[90] M. Ruivo, A. F. Guarda, and F. Pereira, "Double-deep learning-based point cloud geometry coding with adaptive super-resolution," in *2022 10th European Workshop on Visual Information Processing (EUVIP)*. IEEE, 2022, pp. 1–6.

[91] G. Liu, J. Wang, D. Ding, and Z. Ma, "PCGFormer: Lossy point cloud geometry compression via local self-attention," in *2022 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 2022, pp. 1–5.

[92] T. Huang, J. Zhang, J. Chen, Z. Ding, Y. Tai, Z. Zhang, C. Wang, and Y. Liu, "3QNet: 3D point cloud geometry quantization compression network," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–13, 2022.

[93] H. Zheng, W. Gao, Z. Yu, T. Zhao, and G. Li, "ViewPCGC: View-guided learned point cloud geometry compression," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 7152–7161.

[94] Y. Ye and W. Gao, "LLM-PCGC: Large language model-based point cloud geometry compression," *arXiv preprint arXiv:2408.08682*, 2024.

[95] D. T. Nguyen and A. Kaup, "Lossless point cloud geometry and attribute compression using a learned conditional probability model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 4337–4348, 2023.

[96] Y. Li, C. Madarasingha, and K. Thilakarathna, "DiffPMAE: Diffusion masked autoencoders for point cloud reconstruction," in *European Conference on Computer Vision*. Springer, 2024, pp. 362–380.

[97] S. Biswas, J. Liu, K. Wong, S. Wang, and R. Urtasun, "MUSCLE: Multi sweep compression of LiDAR using deep entropy models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 170–22 181, 2020.

[98] Y. Hu, R. Gong, T. Fan, and Y. Wang, "TESO: Representing and compressing 3D point cloud scenes with textured surfel octree," *arXiv preprint arXiv:2508.07083*, 2025.

[99] Z. Hu, M. Zhen, X. Bai, H. Fu, and C.-l. Tai, "JSENET: Joint semantic segmentation and edge detection network for 3D point clouds," in *European conference on computer vision*. Springer, 2020, pp. 222–239.

[100] K. Zhou, Z. Tan, H. Wang, Y.-l. Li, and S. Wang, "Preserving topological and geometric embeddings for point cloud recovery," *arXiv preprint arXiv:2507.19121*, 2025.

[101] C.-E. Himeur, T. Lejemble, T. Pellegrini, M. Paulin, L. Barthe, and N. Mellado, "PCEDNet: A lightweight neural network for fast and interactive edge detection in 3D point clouds," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 1, pp. 1–21, 2021.

[102] M. Loizou, M. Averkiou, and E. Kalogerakis, "Learning part boundaries from 3D point clouds," in *Computer Graphics Forum*, vol. 39, no. 5. Wiley Online Library, 2020, pp. 183–195.

[103] H. A. Arief, M. Arief, M. Bhat, U. G. Indahl, H. Tveite, and D. Zhao, "Density-adaptive sampling for heterogeneous point cloud object segmentation in autonomous vehicle applications." in *CVPR Workshops*, 2019, pp. 26–33.

[104] J. Kim, G. Bang, K. Choi, M. Seong, J. Yoo, E. Pyo, and J. W. Choi, "Pillargen: Enhancing radar point cloud density and quality via pillar-based point generation network," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 9117–9124.

[105] A. Hojjat, J. Haberer, and O. Landsiedel, "ProgDTD: Progressive learned image compression with double-tail-drop training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1130–1139.

[106] T. Wang, F. Hao, Q. Zhang, and J. Cheng, "Progressive background-foreground difference enhancement for few-shot 3D point cloud semantic segmentation," *Image and Vision Computing*, p. 105656, 2025.

[107] X. Li, J. Lu, H. Ding, C. Sun, J. T. Zhou, and Y. M. Chee, "PointCVaR: Risk-optimized outlier removal for robust 3D point cloud classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 19, 2024, pp. 21 340–21 348.

[108] A. Presta, E. Tartaglione, A. Fiandrotti, M. Grangetto, and P. Cosman, "Efficient progressive image compression with variance-aware masking," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2025, pp. 7692–7700.

[109] Z. Lei, P. Duan, X. Hong, J. F. Mota, J. Shi, and C.-X. Wang, "Progressive deep image compression for hybrid contexts of image classification and reconstruction," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 72–89, 2022.

[110] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3D semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1534–1543.

[111] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NUSCENES: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 621–11 631.

[112] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2446–2454.

[113] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "JPEG pleno database: 8i voxelized full bodies (8iVFB v2)-a dynamic voxelized point cloud dataset," 2017.

[114] V. Zyrianov, H. Che, Z. Liu, and S. Wang, "LiDARDM: Generative LiDAR simulation in a generated world," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 6055–6062.

[115] Z. Chen, Z. Qian, S. Wang, and Q. Chen, "Point cloud compression with sibling context and surface priors," in *European Conference on Computer Vision*. Springer, 2022, pp. 744–759.

[116] S. Perry, "JPEG pleno point cloud coding common test conditions v3. 6. ISO," IEC JTC1/SC29/WG1 N, 91058, Tech. Rep., 2021.

[117] C. Wu, J. Zheng, J. Pfrommer, and J. Beyerer, "Attention-based point cloud edge sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5333–5343.

[118] F. Lin, Y. Yue, Z. Zhang, S. Hou, K. Yamada, V. Kolachalama, and V. Saligrama, "INFOCD: a contrastive chamfer distance loss for point cloud completion," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[119] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 605–613.

[120] A. Javaheri, C. Brites, F. Pereira, and J. Ascenso, "Improving PSNR-based quality metrics performance for point cloud geometry," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3438–3442.

[121] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *ITU SG16 Doc. VCEG-M33*, 2001.

[122] Y. Rao, J. Lu, and J. Zhou, "Global-local bidirectional reasoning for unsupervised representation learning of 3D point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5376–5385.

[123] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, "Attention mechanisms in computer vision: A survey," *Computational Visual Media*, vol. 8, no. 3, pp. 331–368, 2022.

[124] X. Wang, Y. Xu, K. Xu, A. Tagliasacchi, B. Zhou, A. Mahdavi-Amiri, and H. Zhang, "Pie-Net: Parametric inference of point cloud edges," *Advances in neural information processing systems*, vol. 33, pp. 20 167–20 178, 2020.

[125] Y. Liu, L. Zhu, H. Ye, S. Huang, X. Gao, X. Zheng, and S. Shen, "BWFormer: Building wireframe reconstruction from airborne LiDAR point cloud with transformer," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 22 215–22 224.

[126] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "EC-Net: an edge-aware point set consolidation network," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 386–402.

[127] J.-H. Lee, S. Jeon, K. P. Choi, Y. Park, and C.-S. Kim, "DPICT: Deep progressive image compression using trit-planes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 113–16 122.

[128] Y. Chen, W. Zhang, D. Li, J. Wang, and G. Li, "Hierarchical attention networks for lossless point cloud attribute compression," *arXiv preprint arXiv:2504.00481*, 2025.

[129] Z. Luo, C. Zhou, L. Pan, G. Zhang, T. Liu, Y. Luo, H. Zhao, Z. Liu, and S. Lu, "Exploring point-BEV fusion for 3D point cloud object tracking with transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 9, pp. 5921–5935, 2024.

[130] C. He, R. Li, G. Zhang, and L. Zhang, "Scatterformer: Efficient voxel transformer with scattered linear attention," in *European Conference on Computer Vision*. Springer, 2024, pp. 74–92.

[131] D. Bazazian and M. E. Parés, "EDC-Net: Edge detection capsule network for 3D point clouds," *Applied Sciences*, vol. 11, no. 4, p. 1833, 2021.

[132] W. Shen, B. Zhang, H. Xu, X. Li, and J. Wu, "Multi-space point geometry compression with progressive relation-aware transformer," *IEEE Transactions on Multimedia*, vol. 26, pp. 8969–8980, 2024.

[133] C. Li, P. Zhang, and Y. Wu, "Density-aware global-local attention network for point cloud segmentation," *arXiv preprint arXiv:2412.00489*, 2024.

[134] F. Afsana, M. Paul, F. Tohidi, and P. Gao, "A density-aware point cloud geometry compression leveraging cluster-centric processing," *IEEE Access*, vol. 12, pp. 81 441–81 452, 2024.

[135] R. Mekuria and L. Bivolarsky, "Overview of the MPEG activity on point cloud compression," in *2016 Data Compression Conference (DCC)*. IEEE, 2016, pp. 620–620.

[136] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image

compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001.

[137] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[138] K. You, P. Gao, and Q. Li, "IPDAE: Improved patch-based deep autoencoder for lossy point cloud geometry compression," in *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis*, 2022, pp. 1–10.

[139] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 259–16 268.

[140] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional shape-contextnet for point cloud recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Conference*, 2018, pp. 4606–4615.

[141] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-Bert: Pre-training 3D point cloud transformers with masked point modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 313–19 322.

[142] D. Lu, K. Gao, Q. Xie, L. Xu, and J. Li, "3DPCT: 3D point cloud transformer with dual self-attention," *arXiv preprint arXiv:2209.11255*, vol. 1, no. 2, 2022.

[143] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "Pointweb: Enhancing local neighborhood features for point cloud processing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5565–5573.

[144] F. Lin, Y. Yue, S. Hou, X. Yu, Y. Xu, K. D. Yamada, and Z. Zhang, "Hyperbolic chamfer distance for point cloud completion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 595–14 606.

[145] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Learning semantic segmentation of large-scale point clouds with random sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8338–8354, 2021.

[146] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 330–33 342, 2022.

[147] C. Wen, B. Yu, and D. Tao, "Learnable skeleton-aware 3D point cloud sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 671–17 681.

[148] Y. Lin, Y. Huang, S. Zhou, M. Jiang, T. Wang, and Y. Lei, "DA-Net: Density-adaptive downsampling network for point cloud classification via end-to-end learning," in *2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*. IEEE, 2021, pp. 13–18.

[149] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.

[150] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang, "Edge-aware point set resampling," *ACM transactions on graphics (TOG)*, vol. 32, no. 1, pp. 1–12, 2013.

[151] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proceedings of the IEEE/CVF Conference on Computer*

Vision and Pattern Recognition, 2019, pp. 9621–9630.

[152] W. Wu, L. Fuxin, and Q. Shan, "PointConvFormer: Revenge of the point-based convolution," in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition*, 2023, pp. 21 802–21 813.

[153] Z. Li, X. Tang, Z. Xu, X. Wang, H. Yu, M. Chen *et al.*, "Geodesic self-attention for 3D point clouds," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6190–6203, 2022.

[154] H. Fan, Y. Yang, and M. Kankanhalli, "Point 4D transformer networks for spatio-temporal modeling in point cloud videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 204–14 213.

[155] A. Seleem, A. F. Guarda, N. M. Rodrigues, and F. Pereira, "Deep learning-based compressed domain point cloud classification," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 2620–2624.

[156] T. Le and Y. Duan, "PointGrid: A deep network for 3D shape understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9204–9214.

[157] B. Liu, S. Li, X. Sheng, L. Li, and D. Liu, "Joint optimized point cloud compression for 3D object detection," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 1185–1189.

[158] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 604–621.

[159] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia, "Stratified transformer for 3D point cloud segmentation," in *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8500–8509.

[160] G. Zhang, W. Zhao, J. Liu, and X. Liu, "REPS: Reconstruction-based point cloud sampling," *arXiv preprint arXiv:2403.05047*, 2024.

[161] W. Liu, J. Sun, W. Li, T. Hu, and P. Wang, "Deep learning on point clouds and its application: A survey," *Sensors*, vol. 19, no. 19, p. 4188, 2019.

[162] O. Dovrat, I. Lang, and S. Avidan, "Learning to sample," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2760–2769.

[163] X. Wang, Y. Jin, Y. Cen, C. Lang, and Y. Li, "Pst-Net: Point cloud sampling via point-based transformer," in *Image and Graphics: 11th International Conference, ICIG 2021, Haikou, China, August 6–8, 2021, Proceedings, Part III*. Springer, 2021, pp. 57–69.

[164] D. Li, Y. Wei, and R. Zhu, "A comparative study on point cloud down-sampling strategies for deep learning-based crop organ segmentation," *Plant Methods*, vol. 19, no. 1, p. 124, 2023.

[165] H. Zu, J. Zhu, X. Wang, X. Zhang, N. Chen, G. Guo, and Z. Chen, "Research on self-adaptive grid point cloud down-sampling method based on plane fitting and mahalanobis distance gaussian weighting," *Neurocomputing*, vol. 634, p. 129746, 2025.

[166] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *Proceedings of the Annual Allerton Conference on Communication, Control and Computing*, 2000.

[167] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational

image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[168] G. Bjontegaard, "Improvements of the BD-PSNR model," *VCEG-AI11*, 2008.

[169] K. Liu, K. You, P. Gao, and M. Paul, "Att 2 CPC: Attention-guided lossy attribute compression of point clouds," *IEEE Transactions on Artificial Intelligence*, 2024.

[170] T. Koike-Akino and Y. Wang, "Stochastic bottleneck: Rateless auto-encoder for flexible dimensionality reduction," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 2735–2740.

[171] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimization of nonlinear transform codes for perceptual quality," in *2016 Picture Coding Symposium (PCS)*. IEEE, 2016, pp. 1–5.

[172] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[173] C. Cai, L. Chen, X. Zhang, G. Lu, and Z. Gao, "A novel deep progressive image compression framework," in *2019 Picture Coding Symposium (PCS)*. IEEE, 2019, pp. 1–5.

[174] E. Diao, J. Ding, and V. Tarokh, "Drasic: Distributed recurrent autoencoder for scalable image compression," in *2020 Data Compression Conference (DCC)*. IEEE, 2020, pp. 3–12.

[175] K. Islam, L. M. Dang, S. Lee, and H. Moon, "Image compression with recurrent neural network and generalized divisive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1875–1879.

[176] N. Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, S. J. Hwang, J. Shor, and G. Toderici, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4385–4393.

[177] C.-H. Huang and J.-L. Wu, "Unveiling the future of human and machine coding: A survey of end-to-end learned image compression," *Entropy*, vol. 26, no. 5, p. 357, 2024.

[178] Y. Lu, Y. Zhu, Y. Yang, A. Said, and T. S. Cohen, "Progressive neural image compression with nested quantization and latent ordering," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 539–543.

[179] S. Li, H. Li, W. Dai, C. Li, J. Zou, and H. Xiong, "Learned progressive image compression with dead-zone quantizers," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 6, pp. 2962–2978, 2022.

[180] G. Vadivel, S.-C. Cheng, C.-C. Huang, and J.-Y. Su, "Progressive point cloud compression with the fusion of symmetry based convolutional neural pyramid and vector quantization," in *2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. IEEE, 2021, pp. 1–2.

[181] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[182] F. Galligan, M. Hemmer, O. Stava, F. Zhang, and J. Brettle, "Google/Draco: a library for compressing and decompressing 3D geometric meshes and point

clouds," *Draco: A Library for Compressing and Decompressing 3D Geometric Meshes and Point clouds*, 2018.

[183] R. Mekuria, K. Blom, and P. Cesar, "Design, implementation, and evaluation of a point cloud codec for tele-immersive video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 828–842, 2016.