Article

Non-Uniform Voxelisation for Point Cloud Compression

Bert Van Hauwermeiren *, Leon Denis, Adrian Munteanu *

Posted Date: 13 January 2025

doi: 10.20944/preprints202501.0853.v1

Keywords: point cloud; compression; quantisation; voxelisation

Article

Non-Uniform Voxelisation for Point Cloud Compression

Bert Van Hauwermeiren *, Leon Denis and Adrian Munteanu *

Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
* Correspondence: bert.karel.van.hauwermeiren@vub.be, adrian.Munteanu@vub.be

Abstract: Point cloud compression is essential for the efficient storage and transmission of 3D data in various applications, such as virtual reality, autonomous driving, and 3D modelling. Most existing compression methods employ voxelisation, all of which uniform, to partition 3D space into voxels for more efficient compression. However, uniform voxelisation may not capture the underlying geometry of complex scenes effectively. In this paper, we propose a novel non-uniform voxelisation technique for point cloud geometry compression. Our method adaptively adjusts voxel sizes based on local point density, preserving geometric details while enabling more accurate reconstructions. Through comprehensive experiments on well-known benchmark datasets, ScanNet and ModelNet, we demonstrate that our approach achieves better compression ratios and reconstruction quality in comparison to traditional uniform voxelisation methods. The results highlight the potential of non-uniform voxelisation as a viable and effective alternative, offering improved performance for point cloud geometry compression in a wide range of real-world scenarios.

Keywords: point cloud; compression; quantisation; voxelisation

1. Introduction

In recent years, the demand for point cloud processing techniques has surged, driven by their increasing application in emerging fields such as augmented and virtual reality (AR/VR), autonomous driving and robotics. These applications often rely on the storage and transmission of large point clouds, which typically consist of millions of individual points. Consequently, the development of efficient point cloud compression methods has become critical to addressing the substantial data demands inherent in these systems. Unlike the structured nature of 2D image and video data, 3D point clouds are inherently unstructured, presenting unique challenges in compression. This complexity highlights the need for specialised algorithms and has driven extensive research efforts aimed at optimizing point cloud compression techniques.

Nearly all point cloud compression methods in literature begin by voxelising the input point cloud, converting it into a grid of discrete, integer-coordinated voxels. This voxelisation process not only imposes structure on the otherwise unstructured point cloud but also makes the data more amenable to compression techniques, such as convolution-based approaches and octree partitioning schemes. By introducing a structured representation, voxelisation enables compression algorithms to more effectively exploit spatial redundancies and patterns within the data.

The most established point cloud compression codecs are being developed by the Moving Picture Experts Group (MPEG), the organisation responsible for widely adopted video coding standards such as AVC [1], HEVC [2], and VVC [3]. They are maintaining two distinct point cloud compression standards: the geometry-based point cloud compression standard (G-PCC) [4] and the video-based point cloud compression standard (V-PCC) [5]. The V-PCC standard leverages the power of established video codecs by projecting the 3D point cloud onto 2D planes and then compressing these projections as video streams. This approach benefits from the maturity and optimisation of existing video compression technologies. The G-PCC standard, on the other hand, works purely in 3D and has two

 doi:10.20944/preprints202501.0853.v1

variants. The first option is an octree-based codec, which recursively splits the bounding box around the voxelised point cloud into eight subnodes until the nodes are voxel-sized; at this point, the octree leaves are all the occupied voxels. The only information to be encoded are the occupancy bits, which indicate which subnodes are occupied. So, most active research investigates how to efficiently entropy code these using predictions from recently coded or neighbouring nodes. The octree-based codec recursively divides the bounding box of the voxelised point cloud into eight subnodes, continuing until the nodes become voxel-sized. At this point, the octree leaves represent the occupied voxels, and the compression task is reduced to encoding the occupancy bits, which indicate for each node what subnodes are occupied. Active research in this area focuses on enhancing the efficiency of entropy coding these occupancy bits by leveraging predictions from previously coded or spatially neighboring nodes.

The second approach, which performs predictive geometry coding, is designed for scenarios requiring low-complexity decoding and minimal latency, such as streaming applications. This method uses a prediction graph to model the point cloud, where each point's geometry is estimated based on its predecessors in the graph. This predictive framework enables streamlined compression while maintaining efficient decoding.

Most point cloud compression research can also be classified into geometry-based [6] and video-based codecs [7,8]. The focus of this paper is on voxelisation which is used in geometry-based codecs. Recent geometry-based point cloud compression research has increasingly focused on learning-based approaches, which can typically be grouped into four main categories. Voxel-based methods process voxelised point clouds directly using 3D convolutions, either in dense [9–11] or sparse forms [12–14], with the latter accounting for the inherent sparsity of point clouds. Octree-based methods build upon the octree structure used in the G-PCC standard but enhance it with machine learning to improve entropy coding of occupancy information. For instance, VoxelContext [15] employs convolutional neural networks to predict occupancy probabilities, while transformer-based techniques such as Octattention [16], Octformer [17], and EHEM [18] model complex dependencies between octree nodes to boost compression efficiency. Hybrid methods combine elements of voxel-based and octree-based approaches. For instance, VoxelDNN [19,20] employs a pruned octree structure and encodes its leaves using a voxel-based approach. Finally, some methods operate directly on the unvoxelised version of the point cloud. Examples include approaches based on PointNet++ [21], such as those proposed in [22,23], or methods employing recurrent neural networks [24]. However, these techniques often struggle to scale effectively to higher bitrates and larger point clouds [12,25].

To the best of our knowledge, all existing point cloud compression methods operating on voxelised data rely on uniform voxelisation. However, this approach fails to consider the spatial distribution of points within the input data. As a result, bitrate is often misallocated—excessively spent in sparse regions while insufficiently invested in dense, detail-rich areas. To address these shortcomings, we propose a novel non-uniform voxelisation method that adapts to the specific point distribution of the input cloud, optimizing the allocation of resources to minimise Euclidean coding error. Our approach is compatible with any point cloud codec designed for voxelised data. By demonstrating the substantial benefits of non-uniform voxelisation compared to the traditional uniform approach, this work highlights the need to reconsider voxelisation strategies for advancing the state of point cloud compression.

- We devised a novel approach that allows for non-uniform voxelisation in point cloud compression. The obtained method is based on Lloyd-Max quantisation and minimises Euclidian error.
- We empirically prove the potential of these approaches by a thorough testing on two well-known public datasets: ScanNet [26] and ModelNet10 [27].

The remainder is structured as follows: first, the necessary basics of quantisation are explained in Section 2. Then the proposed methodology is explained in detail in Section 3. The experimental setups and their results are summarised in Section 4. Finally, conclusions are drawn in Section 5.

 doi:10.20944/preprints202501.0853.v1

2. Background Knowledge

Voxelisation of a point cloud is quantising the positions of the points along the three spatial axes. Quantisation is the process of mapping a continuous range of values to a discrete set of symbols. Formally, a quantiser is characterised by its decision boundaries $b_{i\,i=0}^{N}$, where $b_0 = -\infty$ and $b_n = +\infty$, with consecutive pairs of boundaries defining bins that are associated with reconstruction values $y_{i\,i=1}^{N}$. So, the quantisation function can be expressed as

$$Q(x) = y_i \iff b_{i-1} < x \leq b_i. \tag{1}$$

The goal of designing a quantiser is to minimise bitrate while also minimising the distortion introduced in the process. This goal is often formalised by using a lagrangian multiplier λ and minimising $D + \lambda R$. Where the distortion is usually given by the mean squared error

$$\begin{aligned} D &= \mathbb{E}[(x - Q(x))^2] \\ &= \int_{-\infty}^{\infty} (x - Q(x))^2 f(x) dx \\ &= \sum_{k=1}^{M} \int_{b_{k-1}}^{b_k} (x - \hat{x_k})^2 f(x) dx. \end{aligned} \tag{2}$$

While the rate is given by

$$R = \sum_{k=1}^{M} length(c_k) \int_{b_{k-1}}^{b_k} f(x) dx, \tag{3}$$

where c_k is the codeword that is assigned to quantisation bin k. For a sufficiently efficient entropy codec, the cost of each codeword approaches the information content $length(c_k) \approx -log(p_k)$, so the total rate approaches the entropy, $H = -\sum_{k=1}^{M} p_k \log(p_k)$.

In this work, we look at two quantisers:

1. Uniform
   The uniform quantiser is popular due to its simplicity, and it is shown to be optimal for some distributions. Given a series of values in the range $[X_{min}, X_{max}]$ the reconstruction values are given by

   $$y_i = X_{min} + \frac{2i - 1}{2n}(X_{max} - X_{min}) \tag{4}$$

   where N is the number of quantisation levels. The decision boundaries are then in the middle of the adjacent reconstruction levels,

   $$b_i = \frac{y_i + y_{i+1}}{2}. \tag{5}$$

2. Lloyd-Max
   The Lloyd-Max quantiser adapts to the distribution of the input data $f(x)$. When λ in the optimisation problem is set to 0, the problem becomes minimising the mean squared error, which can be solved:

   $$\frac{\partial D}{\partial b_k} = 0 \implies b_k = \frac{y_k + y_{k+1}}{2} \tag{6}$$

   and

   $$\frac{\partial D}{\partial y_k} = 0 \implies y_k = \frac{\int_{b_{k-1}}^{b_k} x f(x) dx}{\int_{b_{k-1}}^{b_k} f(x) dx}. \tag{7}$$

 doi:10.20944/preprints202501.0853.v1

These formulas can be used in an iterative manner. We start with a uniform quantiser, then iteratively update the boundaries and reconstruction levels according to these rules until the average variance in the bins no longer decreases.

3. Proposed Methodology

We propose a novel non-uniform voxelisation method designed to minimise Euclidean distortion, as illustrated in Figure 1. Our approach extends the traditional Lloyd-Max algorithm to efficiently compress point cloud data by applying it independently to the distribution of points across each of the three dimensions. This allows the resulting voxels to take on generalised cuboid shapes, rather than being constrained to cubes. It can be shown that minimising Euclidean error in three-dimensional space reduces to minimising squared error along each dimension independently, assuming decision boundary planes that are orthogonal to the axes.

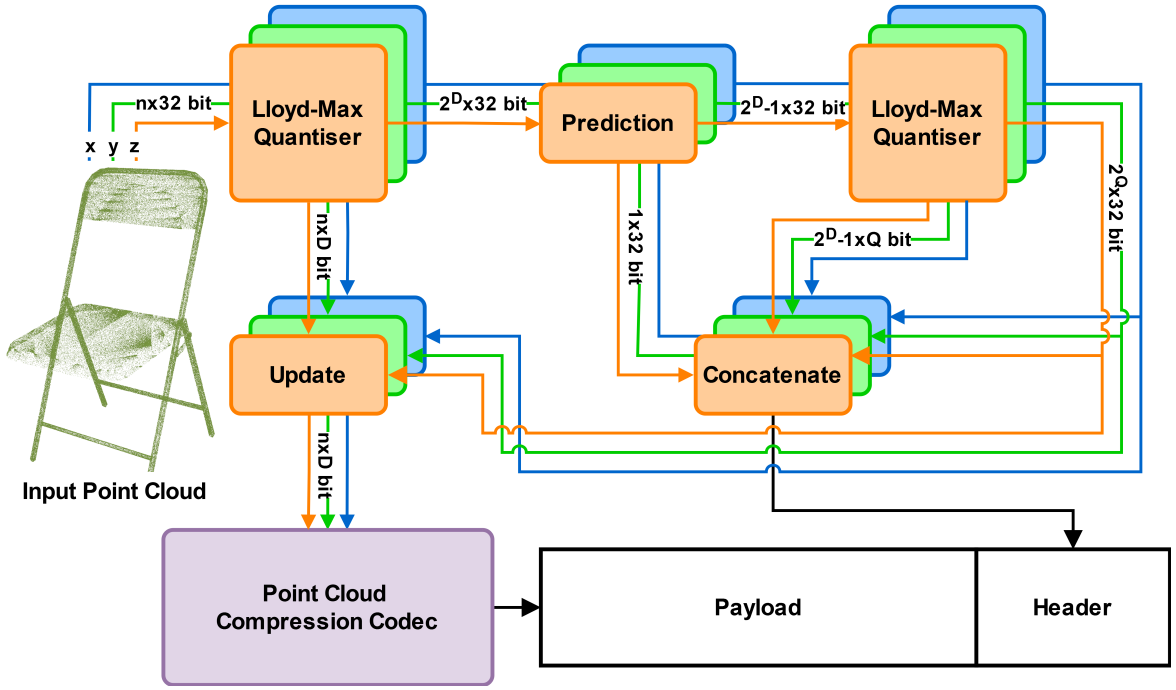


Figure 1. Overview of the proposed method. An input-specific voxelisation is decided by applying Lloyd-Max quantisation for each of the input dimensions. Subsequently, another round of Lloyd-Max quantisation is applied to the reconstruction values after a linear prediction in order to reduce their overhead dramatically. Finally, the input point cloud is voxelised using the determined quantisers, ready for compression using any voxel- or octree-based codec. The quantiser information, which is necessary for the calculation of reconstruction values, is stored in the header. All denoted bitrate sizes are for a singular dimension.

 doi:10.20944/preprints202501.0853.v1

$$
\begin{aligned}
D_{X,Y,Z} &= \mathbb{E}[d(P,Q(P))^2] \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} d(P,Q(P))^2 f(x,y,z)dxdydz \\
&= \sum_{k_1=1}^{N}\sum_{k_2=1}^{N}\sum_{k_3=1}^{N}\int_{b_{k_1-1}}^{b_{k_1}}\int_{b_{k_2-1}}^{b_{k_2}}\int_{b_{k_3-1}}^{b_{k_3}} \sqrt{(x-\hat{x}_{k_1})^2+(y-\hat{y}_{k_2})^2+(z-\hat{z}_{k_3})^2}^2 f(x,y,z)dxdydz \\
&= \sum_{k_1=1}^{N}\sum_{k_2=1}^{N}\sum_{k_3=1}^{N}\int_{b_{k_1-1}}^{b_{k_1}}\int_{b_{k_2-1}}^{b_{k_2}}\int_{b_{k_3-1}}^{b_{k_3}} (x-\hat{x}_{k_i})^2 f(x,y,z)dxdydz + \dots \\
&= \sum_{k_1=1}^{N}\int_{b_{k_1-1}}^{b_{k_1}}\left(\sum_{k_2=1}^{N}\sum_{k_3=1}^{N}\int_{b_{k_2-1}}^{b_{k_2}}\int_{b_{k_3-1}}^{b_{k_3}} (x-\hat{x}_{k_i})^2 f(x,y,z)dydz\right)dx + \dots \\
&= \sum_{k_1=1}^{N}\int_{b_{k_1-1}}^{b_{k_1}} (x-\hat{x}_{k_i})^2 f(x)dx + \dots \\
&= D_X + D_Y + D_Z.
\end{aligned}
\tag{8}
$$

While these assumptions may appear restrictive, they are crucial for the practical application of non-uniform quantisation to point clouds. Although a vector quantisation approach might initially appear more suitable due to its ability to perform generalised multi-dimensional quantisation, voxelisation necessitates the creation of an $N \times N \times N$ grid, which vector quantisation is not designed for. Additionally, the reconstruction values must be included in the geometry header to ensure accurate point cloud reconstruction on the decoder side. Making the quantisers interdependent introduces a size complexity of N^3 for the header overhead, which is impractical for real-world applications. As we will demonstrate, even the $3N$ complexity of independent quantisers poses significant challenges. Finally, optimising and encoding non-planar quantisation boundaries is beyond the scope of this work.

Another important consideration is the execution time of the voxelisation process. While extracting the point distribution and determining an appropriate quantiser along each axis is straightforward and efficient, directly processing the points in three-dimensional space is far more complex. These complexities are a key reason why most existing approaches incorporate a voxelisation step in the first place.

To accurately reconstruct the point cloud at the decoder, it is necessary to encode the mapping between the internal integer coordinate system and the actual point cloud coordinates. However, the bitrate required is very substantial due to the large number of voxels along each dimension, making it impractical to transmit this data directly. As the octree depth increases, the number of reconstruction levels grows exponentially, with each level requiring a 32-bit float representation. At higher quality levels, the overhead of transmitting this information becomes prohibitive. Figure 2 illustrates the considerable overhead introduced by communicating the reconstruction levels compared to just the outer boundaries of a uniform approach. The impact of header overhead on the efficiency of the proposed method will be further explored in Section 4.3.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 January 2025 doi:10.20944/preprints202501.0853.v1

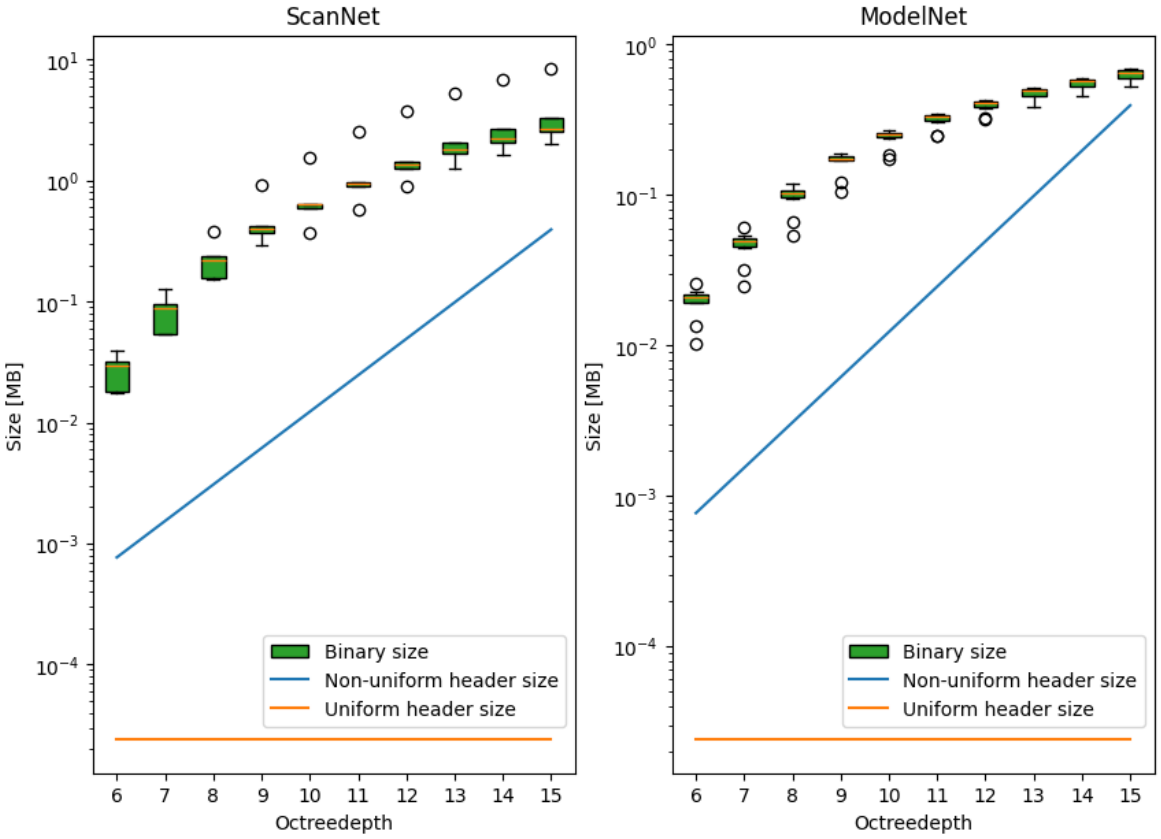


Figure 2. Illustration of the large bitrate overhead of a naive non-uniform approach for the ScanNet and ModelNet datasets. The increased overhead would quickly nullify the benefit of any increase in compression performance. The point clouds are compressed using uniform voxelisation and G-PCC.

To address the overhead, our approach applies an additional Lloyd-Max quantiser to the reconstruction levels in order to reduce the introduced overhead. By quantising the output reconstruction levels of the Lloyd-Max quantiser, we obtain an approximate version of the original quantiser. The primary goal is to minimise the additional distortion, denoted as $\hat{D}$, introduced by this approximation, relative to the optimal distortion achieved by the original quantiser. It is important to note that only the reconstruction values need to be communicated and approximated, meaning both quantisers share the same boundaries.

$$\begin{aligned}
\hat{D} &= \mathbb{E}[(x-\hat{Q}(x))^2] - \mathbb{E}[(x-Q(x))^2] \\
&= \mathbb{E}[\hat{Q}(x)^2 - Q(x)^2] - 2\,\mathbb{E}[x\hat{Q}(x) - xQ(x)] \\
&= \sum_{n=1}^{N} (\hat{y}_n^2 - y_n^2) \int_{b_{n-1}}^{b_n} f(x)dx - 2\sum_{n=1}^{N} (\hat{y}_n - y_n) \int_{b_{n-1}}^{b_n} xf(x)dx \\
&= \sum_{n=1}^{N} (\hat{y}_n^2 - y_n^2) p_n - 2\sum_{n=1}^{N} (\hat{y}_n - y_n) p_n y_n \\
&= \sum_{n=1}^{N} (\hat{y}_n - y_n)^2 p_n,
\end{aligned} \tag{9}$$

where p_n is the probability or importance of a certain interval. We apply quantisation on the reconstruction levels only after a prediction step to more efficiently spend the quantisation bits. Thus, the distortion to be minimised by the second Lloyd-Max quantiser is a discrete version of equation 2:

$$\hat{D} = \sum_{k=1}^{K} \sum_{\{n|b_{k-1}<d_n<b_k\}} (d_n - \hat{d}_k)^2 p_n, \tag{10}$$

 doi:10.20944/preprints202501.0853.v1

where $\hat{d}_k$ are the reconstruction values of the second quantiser and $d_n = y_n - y_{n-1}$.

4. Results

4.1. Datasets

The efficiency of the proposed method is demonstrated through performance evaluation on two widely used public datasets. ScanNet [26] is a popular 3D indoor scene point cloud dataset commonly used for tasks such as segmentation, compression, and other point cloud processing applications. We evaluate the proposed method using 20 scenes from this dataset. The second dataset, ModelNet [28], consists of synthetic meshes representing various object categories. For evaluation, we selected two meshes from each of the ten categories, sampling 1,000,000 points and their associated normals from each mesh.

4.2. Experimental Details

All voxelisation in the state-of-the-art point cloud compression research is uniform, which thus serves as the baseline for comparison for our proposed non-uniform voxelisation approach. To measure the effect of the proposed method on bitrate, we encode the voxelised point cloud using the G-PCC standard [4], which is a state-of-the-art point cloud compression codec that is widely used as a benchmark in the point cloud compression field. Rate-distortion tuning is achieved by adjusting the octree depth parameter, which in our experiments varies between 6 and 16. To evaluate performance, we use four key metrics. The bitrate is measured in bits per point (bpp). Reconstruction quality is assessed by calculating the PSNR of the point-to-point and point-to-plane distortions, as defined in [29], with $p = 1$. Additionally, the Bjontegaard Delta rate (BD-rate) is employed to quantify overall gains in rate and quality. All experiments are performed on a desktop system with an Intel Core i9-9900K CPU and 64 GB of memory. A new parameter introduced by the proposed method is the number of bits used for the reconstruction values in the second quantisation, referred to as Q-bits. Lower values result in significant rate savings, while higher values yield higher-quality reconstructions. Empirical results indicate that setting this parameter to four provides the optimal performance, as detailed in Section 4.3.

4.3. Q-Bits Tuning

In addition to the octree depth, which defines the number of quantisation bits for the first quantiser, the proposed approach introduces a second tunable parameter: the number of quantisation bits for the second quantiser, referred to as Q-bits. We investigated the effects of tuning this parameter and identified optimal values for performance. The results are shown in Table 1. The first notable observation is the importance of the second quantisation, as the method performs significantly worse at 12 bits per reconstruction value compared to the uniform approach. Furthermore, the method exhibits robust performance across a range of smaller Q-bit values, suggesting that fine-tuning this parameter is not required to achieve good results.

Table 1. Quantitative comparison of the coding performance for different Q-bits values, the metric being BD-rate gains (point-to-point and point-to-plane) compared against the uniform approach. The best results are ***highlighted***.

Q-bits	ScanNet		ModelNet	
	D1 BD-rate	D2 BD-rate	D1 BD-rate	D2 BD-rate
2	-11.11%	-5.51%	-5.66%	-25.05%
3	-12.04%	-6.35%	-8.20%	-38.20%
4	***-12.21%***	***-6.51%***	***-8.40%***	-41.49%
6	-11.91%	-6.15%	-8.03%	***-42.49%***
8	-10.76%	-4.90%	-6.82%	-41.71%
12	5.92%	12.61%	9.90%	-28.94%

 doi:10.20944/preprints202501.0853.v1

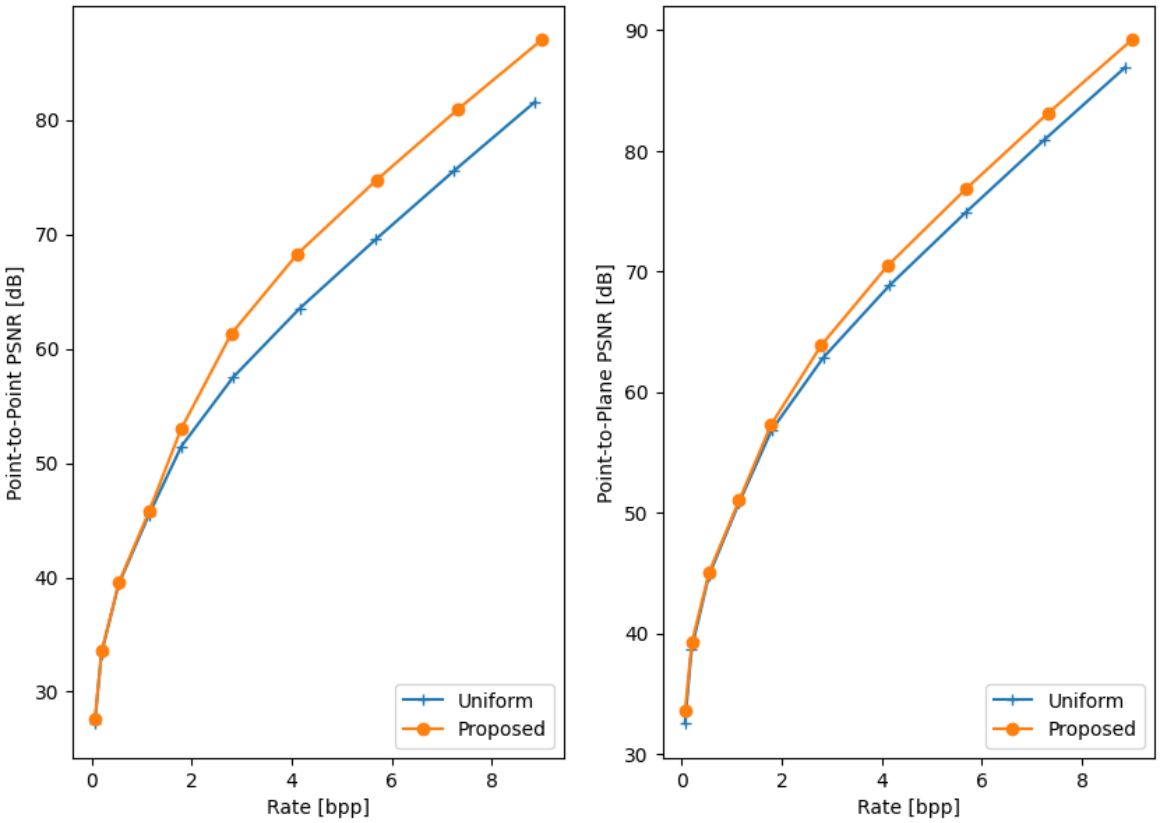
4.4. Comparison to SOTA

The results of the comparison between the proposed method and uniform voxelisation are presented in Table 2 and Figure 3. These results demonstrate the efficacy of the proposed method, with average BD-rate gains of 10.41% and 22.34% for point-to-point and point-to-plane distortion, respectively. The proposed method consistently achieves significantly higher PSNRs while incurring only a slight increase in rate. A qualitative comparison between the proposed method and the baseline is shown in Figure 4. The strength of the non-uniform approach lies in its ability to adapt to the point distribution within the input point cloud, such as aligning the reconstruction values with dense regions like the seat of the ModelNet chair or the floor and walls in the ScanNet scene. In contrast, the uniform approach fails to account for point distribution, often resulting in large errors in these critical regions.

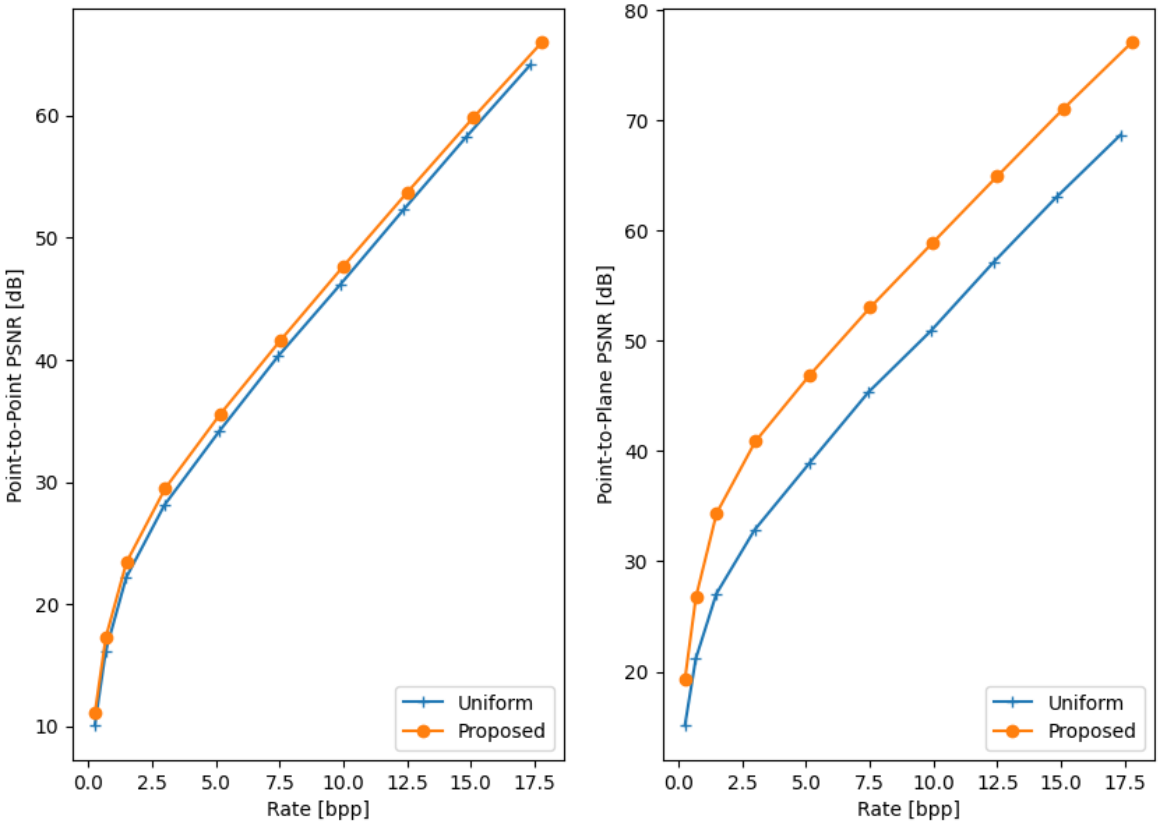
Table 2. Comparison of rate-distortion performance of the proposed method using the BD-rate gain. D1 and D2 refer to point-to-point and point-to-plane distortions, respectively.

Data		Uniform			Proposed			BD-rate gain	
		Rate	D1 PSNR	D2 PSNR	Rate	D1 PSNR	D2 PSNR	D1	D2
Scannet		3.25	54.45	59.80	3.27	57.18	60.98	-12.20	-6.52
ModelNet	bathtub	7.60	36.18	40.73	7.63	37.39	44.85	-8.40	-26.96
	bed	7.41	32.56	37.81	7.53	33.83	44.54	-7.41	-38.93
	chair	5.65	39.79	45.12	5.72	40.98	52.20	-9.52	-46.88
	desk	7.51	34.20	39.13	7.60	35.90	49.61	-10.15	-50.86
	dresser	6.87	37.68	41.59	6.99	39.95	54.68	-13.67	-58.71
	monitor	5.63	45.59	51.54	5.74	46.97	64.25	-8.60	-63.83
	nightstand	7.71	40.19	44.36	7.81	41.74	48.78	-9.45	-26.87
	sofa	7.60	30.15	37.08	7.68	30.79	40.11	-3.64	-20.22
	table	7.61	39.16	43.48	7.75	40.57	47.58	-8.73	-26.17
	toilet	7.61	40.95	45.23	7.70	41.94	48.28	-6.69	-22.13
Average		5.18	46.05	51.20	5.24	48.09	55.23	-10.41	-22.34

 doi:10.20944/preprints202501.0853.v1



(a) ScanNet



(b) ModelNet

Figure 3. Rate-Distorion curves of the proposed method and the baseline on the ScanNet 3a and ModelNet 3b datasets.

 doi:10.20944/preprints202501.0853.v1

(a) ScanNet

(b) ModelNet

Figure 4. A qualitative comparison of uniform voxelisation and the proposed non-uniform voxelisation on a sample from the ScanNet 4a and ModelNet 4b datasets. Each point is coloured from red to green based on the Euclidian coding error (D1) for that point, the colour scale is shared between the methods.

4.5. Impact of Density

Point cloud compression methods are designed to handle a wide range of densities, from sparse point clouds captured by real-time depth sensors to highly dense point clouds used in applications such as heritage preservation. To assess the impact of point cloud density on the performance of the proposed method, we conducted experiments using the ModelNet dataset. Since ModelNet provides mesh data, we sampled it to generate point clouds with varying densities, allowing for controlled experimentation across a broad range of point counts. Specifically, we sampled point clouds containing between 50,000 and 4,000,000 points, which reflects the typical density variations encountered in real-world datasets, as summarised in Table 3.

 doi:10.20944/preprints202501.0853.v1

Table 3. Densities of commonly used datasets for point cloud processing.

Dataset	Average points per frame
ScanNet [26]	3.3M
KITTI [30]	120K
nuScenes [31]	34K
Waymo Open Dataset [32]	177K
S3DIS [33] (Room / Area)	1M / 45M

The results of the density experiment are presented in Table 4. The findings indicate that the proposed method performs better for denser point clouds. This can be attributed to the reduced relative overhead of communicating reconstruction values, which does not scale proportionally with the number of points. The proposed method consistently outperforms the uniform voxelisation approach across all density levels. Additionally, as the density increases, the rate allocated to quantisation becomes a smaller fraction of the total rate, enabling higher Q-bits values without a significant increase in cost. These results highlight the robustness and scalability of the proposed method, making it particularly well-suited for applications that involve high-density point clouds.

Table 4. Performance on point clouds with different densities.

# Points	Q-bits	Uniform			Proposed			BD-rate gain	
		Rate	D1 PSNR	D2 PSNR	Rate	D1 PSNR	D2 PSNR	D1	D2
50 000	2	12.60	37.18	42.00	13.46	39.03	47.62	-3.31	-19.55
100 000	3	11.18	37.18	42.00	11.83	38.97	48.85	-4.76	-27.32
250 000	4	9.49	37.19	42.04	9.85	38.74	49.39	-6.16	-34.21
500 000	4	8.30	37.18	42.02	8.50	38.61	49.37	-7.40	-38.05
1 000 000	4	7.24	37.18	42.02	7.35	38.54	49.33	-8.40	-41.49
4 000 000	5	5.38	37.19	42.04	5.42	38.49	49.58	-10.51	-49.78

4.6. Execution Time

The final aspect of the proposed method to evaluate is its execution time. While the method's adaptability to the input point cloud enhances its performance, it also introduces additional computational complexity. The execution times are provided in Table 5. Despite this increased complexity, the rise in execution time is relatively small compared to the overall processing time of the G-PCC codec. Therefore, the trade-off in execution time is modest when weighed against the significant improvements in coding efficiency offered by the proposed method.

 doi:10.20944/preprints202501.0853.v1

Table 5. Comparison between the execution time of the complete codec using uniform and non-uniform voxelisation. The runtimes are expressed in milliseconds.

octree depth	ScanNet			ModelNet		
	Uniform	Proposed	Δ	Uniform	Proposed	Δ
6	5194.60	5613.99	8.07%	1804.75	2180.99	8.85%
7	5190.60	5508.96	6.13%	1867.25	2059.26	10.28%
8	5143.20	5543.84	7.79%	1909.05	2090.62	9.51%
9	5257.00	5678.42	8.02%	2111.20	2280.34	8.01%
10	5759.20	6108.41	6.06%	2354.25	2513.61	6.77%
11	6593.20	6862.53	4.09%	2603.80	2729.44	4.83%
12	7177.60	7666.90	6.82%	2768.40	2966.85	7.17%
13	7903.00	8410.00	6.42%	2989.00	3183.79	6.52%
14	8644.40	9112.04	5.41%	3161.90	3413.44	7.96%
15	9239.60	9910.39	7.26%	3454.80	3665.64	6.10%

5. Conclusions

This paper presents a novel approach to point cloud compression by introducing non-uniform, distortion-minimizing voxelisation—representing the first exploration of non-uniform quantisation in this domain. Unlike traditional methods that rely on uniform voxelisation, the proposed technique adapts voxel sizes to the local distribution of points, thereby optimizing compression performance. The method's efficiency and robustness were thoroughly validated through extensive experiments on the well-established public datasets, ScanNet and ModelNet. The results highlight the potential for further exploration of innovative non-uniform voxelisation techniques to advance the field of point cloud compression.

Author Contributions: Conceptualization, B.V. and A.M.; methodology, B.V.; software, B.V.; validation, B.V. and A.M.; formal analysis, B.V.; investigation, B.V.; resources, A.M.; data curation, B.V.; writing—original draft preparation, B.V.; writing—review and editing, L.D and A.M.; visualization, B.V.; supervision, A.M.; project administration, X.X.; funding acquisition, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by Innoviris within the research project MUSCLES.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The ScanNet dataset is available at http://www.scan-net.org/, the ModelNet dataset is available at https://modelnet.cs.princeton.edu/.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wiegand, T.; Sullivan, G.; Bjontegaard, G.; Luthra, A. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* **2003**, *13*, 560–576. Conference Name: IEEE Transactions on Circuits and Systems for Video Technology, https://doi.org/10.1109/TCSVT.2003.815165.
2. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology* **2012**, *22*, 1649–1668. Conference Name: IEEE Transactions on Circuits and Systems for Video Technology, https://doi.org/10.1109/TCSVT.2012.2221191.
3. Bross, B.; Wang, Y.K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G.J.; Ohm, J.R. Overview of the Versatile Video Coding (VVC) Standard and its Applications. *IEEE Transactions on Circuits and Systems for Video Technology* **2021**, *31*, 3736–3764. Conference Name: IEEE Transactions on Circuits and Systems for Video Technology, https://doi.org/10.1109/TCSVT.2021.3101953.
4. Zhang, W.; Yang, F.; Xu, Y.; Preda, M. Standardization Status of MPEG Geometry-Based Point Cloud Compression (G-PCC) Edition 2. In Proceedings of the 2024 Picture Coding Symposium (PCS), 2024, pp. 1–5. ISSN: 2472-7822, https://doi.org/10.1109/PCS60826.2024.10566443.

5. Mammou, K.; Kim, J.; Tourapis, A.M.; Podborski, D.; Flynn, D. Video and Subdivision based Mesh Coding. In Proceedings of the 2022 10th European Workshop on Visual Information Processing (EUVIP), 2022, pp. 1–6. ISSN: 2471-8963, https://doi.org/10.1109/EUVIP53989.2022.9922888.
6. Shao, Y.; Gao, W.; Liu, S.; Li, G. Advanced Patch-Based Affine Motion Estimation for Dynamic Point Cloud Geometry Compression. *Sensors* **2024**, *24*, 3142. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute, https://doi.org/10.3390/s24103142.
7. Tohidi, F.; Paul, M.; Ulhaq, A.; Chakraborty, S. Improved Video-Based Point Cloud Compression via Segmentation. *Sensors* **2024**, *24*, 4285. Number: 13 Publisher: Multidisciplinary Digital Publishing Institute, https://doi.org/10.3390/s24134285.
8. Dumic, E.; Bjelopera, A.; Nüchter, A. Dynamic Point Cloud Compression Based on Projections, Surface Reconstruction and Video Compression. *Sensors* **2022**, *22*, 197. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, https://doi.org/10.3390/s22010197.
9. Wang, J.; Zhu, H.; Liu, H.; Ma, Z. Lossy Point Cloud Geometry Compression via End-to-End Learning. *IEEE Transactions on Circuits and Systems for Video Technology* **2021**, *31*, 4909–4923. Conference Name: IEEE Transactions on Circuits and Systems for Video Technology, https://doi.org/10.1109/TCSVT.2021.3051377.
10. Quach, M.; Valenzise, G.; Dufaux, F. Learning Convolutional Transforms for Lossy Point Cloud Geometry Compression. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 4320–4324. arXiv:1903.08548 [cs, eess, stat], https://doi.org/10.1109/ICIP.2019.8803413.
11. Zhuang, L.; Tian, J.; Zhang, Y.; Fang, Z. Variable Rate Point Cloud Geometry Compression Method. *Sensors* **2023**, *23*, 5474. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, https://doi.org/10.3390/s23125474.
12. Wang, J.; Ding, D.; Li, Z.; Feng, X.; Cao, C.; Ma, Z. Sparse Tensor-Based Multiscale Representation for Point Cloud Geometry Compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2023**, *45*, 9055–9071. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, https://doi.org/10.1109/TPAMI.2022.3225816.
13. Deng, J.; An, Y.; Li, T.H.; Liu, S.; Li, G. ScanPCGC: Learning-Based Lossless Point Cloud Geometry Compression using Sequential Slice Representation. In Proceedings of the ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024, pp. 8386–8390. ISSN: 2379-190X, https://doi.org/10.1109/ICASSP48485.2024.10447944.
14. Akhtar, A.; Li, Z.; Van der Auwera, G. Inter-Frame Compression for Dynamic Point Cloud Geometry Coding. *IEEE Transactions on Image Processing* **2024**, *33*, 584–594. Conference Name: IEEE Transactions on Image Processing, https://doi.org/10.1109/TIP.2023.3343096.
15. Que, Z.; Lu, G.; Xu, D. VoxelContext-Net: An Octree Based Framework for Point Cloud Compression. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2021, pp. 6042–6051.
16. Fu, C.; Li, G.; Song, R.; Gao, W.; Liu, S. OctAttention: Octree-Based Large-Scale Contexts Model for Point Cloud Compression. *Proceedings of the AAAI Conference on Artificial Intelligence* **2022**, *36*, 625–633. Number: 1, https://doi.org/10.1609/aaai.v36i1.19942.
17. Cui, M.; Long, J.; Feng, M.; Li, B.; Kai, H. OctFormer: Efficient Octree-Based Transformer for Point Cloud Compression with Local Enhancement. *Proceedings of the AAAI Conference on Artificial Intelligence* **2023**, *37*, 470–478. Number: 1, https://doi.org/10.1609/aaai.v37i1.25121.
18. Song, R.; Fu, C.; Liu, S.; Li, G. Efficient Hierarchical Entropy Model for Learned Point Cloud Compression. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023; pp. 14368–14377. https://doi.org/10.1109/CVPR52729.2023.01381.
19. Nguyen, D.T.; Quach, M.; Valenzise, G.; Duhamel, P. Learning-Based Lossless Compression of 3D Point Cloud Geometry. In Proceedings of the ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 4220–4224. ISSN: 2379-190X, https://doi.org/10.1109/ICASSP39728.2021.9414763.
20. Nguyen, D.T.; Quach, M.; Valenzise, G.; Duhamel, P. Lossless Coding of Point Cloud Geometry Using a Deep Generative Model. *IEEE Transactions on Circuits and Systems for Video Technology* **2021**, *31*, 4617–4629. Conference Name: IEEE Transactions on Circuits and Systems for Video Technology, https://doi.org/10.1109/TCSVT.2021.3100279.
21. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 652–660.

22. You, K.; Gao, P. Patch-Based Deep Autoencoder for Point Cloud Geometry Compression. In Proceedings of the ACM Multimedia Asia, 2021, pp. 1–7. arXiv:2110.09109 [cs], https://doi.org/10.1145/3469877.3490611.
23. Huang, T.; Liu, Y. 3D Point Cloud Geometry Compression on Deep Learning. In Proceedings of the Proceedings of the 27th ACM International Conference on Multimedia, New York, NY, USA, 2019; MM '19, pp. 890–898. https://doi.org/10.1145/3343031.3351061.
24. Tu, C.; Takeuchi, E.; Carballo, A.; Takeda, K. Point Cloud Compression for 3D LiDAR Sensor using Recurrent Neural Network with Residual Blocks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 3274–3280. ISSN: 2577-087X, https://doi.org/10.1109/ICRA.2019.8794264.
25. Quach, M.; Pang, J.; Tian, D.; Valenzise, G.; Dufaux, F. Survey on Deep Learning-Based Point Cloud Compression. *Frontiers in Signal Processing* **2022**, *2*. Publisher: Frontiers, https://doi.org/10.3389/frsip.2022.846972.
26. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Niessner, M. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017; pp. 2432–2443. https://doi.org/10.1109/CVPR.2017.261.
27. Zhirong Wu.; Song, S.; Khosla, A.; Fisher Yu.; Linguang Zhang.; Xiaoou Tang.; Xiao, J. 3D ShapeNets: A deep representation for volumetric shapes. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015; pp. 1912–1920. https://doi.org/10.1109/CVPR.2015.7298801.
28. Maglo, A.; Lavoué, G.; Dupont, F.; Hudelot, C. 3D Mesh Compression: Survey, Comparisons, and Emerging Trends. *ACM Computing Surveys* **2015**, *47*, 44:1–44:41. https://doi.org/10.1145/2693443.
29. Tian, D.; Ochimizu, H.; Feng, C.; Cohen, R.; Vetro, A. Geometric distortion metrics for point cloud compression. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 3460–3464. ISSN: 2381-8549, https://doi.org/10.1109/ICIP.2017.8296925.
30. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361. ISSN: 1063-6919, https://doi.org/10.1109/CVPR.2012.6248074.
31. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A Multimodal Dataset for Autonomous Driving. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020; pp. 11618–11628. https://doi.org/10.1109/CVPR42600.2020.01164.
32. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in Perception for Autonomous Driving: Waymo Open Dataset, 2020. arXiv:1912.04838 [cs].
33. Armeni, I.; Sax, S.; Zamir, A.R.; Savarese, S. Joint 2D-3D-Semantic Data for Indoor Scene Understanding, 2017. arXiv:1702.01105 [cs].