AnyPcc: Compressing Any Point Cloud with a Single Universal Model

Kangli Wang[1*] Qianxi Yi[1,2*] Yuqi Ye[1] Shihao Li[1] Wei Gao[1,2†]
[1] SECE, Peking University [2] Peng Cheng Laboratory
kangliwang@stu.pku.edu.cn , gaowei262@pku.edu.cn
Project Website: anypcc.github.io

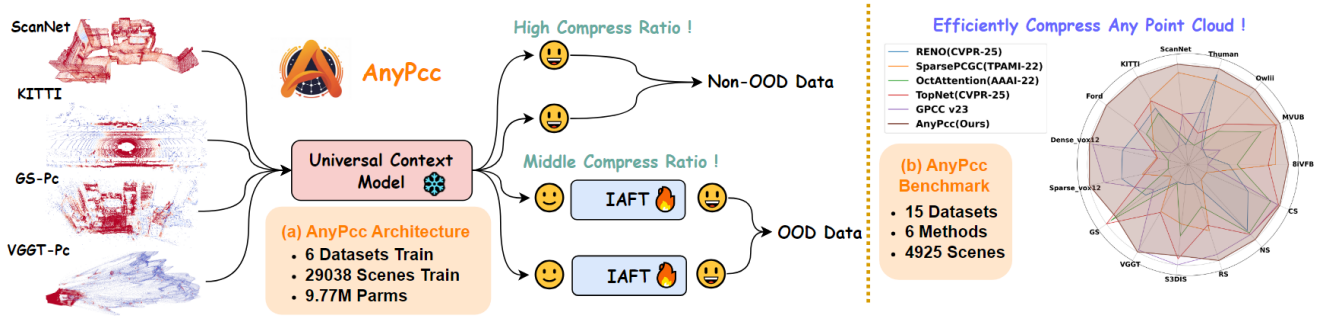

Figure 1. **(a) AnyPcc Architecture.** A single, unified model compresses point clouds from any source, with our Instance-Adaptive Fine-Tuning (IAFT) module boosting performance on out-of-distribution (OOD) data. **(b) AnyPcc Benchmark.** Our comprehensive benchmark features 15 diverse datasets, including both standard and extreme cases. When compared against five state-of-the-art methods, AnyPcc consistently achieves high compression efficiency across all types of point clouds.

Abstract

Generalization remains a critical challenge in deep learning-based point cloud geometry compression. While existing methods perform well on standard benchmarks, their performance collapses in real-world scenarios due to two fundamental limitations: the lack of context models that are robust across diverse data densities, and the inability to efficiently adapt to out-of-distribution (OOD) data. To overcome both challenges, we introduce AnyPcc, a universal point cloud compression framework. AnyPcc first employs a Universal Context Model that leverages coarse-grained spatial priors with fine-grained channel priors to ensure robust context modeling across the entire density spectrum. Second, our novel Instance-Adaptive Fine-Tuning (IAFT) strategy tackles OOD data by synergizing explicit and implicit compression paradigms. For each instance, it fine-tunes a small subset of network weights and transmits them within the bitstream. The minimal bitrate overhead from these weights is significantly outweighed by the resulting gains in geometry compression. Extensive experiments on a benchmark of 15 diverse datasets confirm that AnyPcc sets a new state-of-the-art in point cloud compression while maintaining low complexity. Our code and datasets have been released to encourage reproducible research.

1. Introduction

With the rise of 3D applications like autonomous driving and virtual reality, point clouds have become a standard format for 3D data. This widespread use makes efficient geometry compression essential for reducing storage and transmission costs. While recent learning-based methods outperform traditional standards like G-PCC [5] on certain standard benchmarks, a significant generalization gap often limits their practical use. We argue this gap originates from two fundamental limitations. First, existing context models [52, 54, 66] are often tailored to **specific point cloud densities** and thus fail to maintain stable performance across the wide spectrum of densities found in real-world data, such as sparse LiDAR scans and dense reconstruction outputs. Second, these models **suffer a sharp decline in compression efficiency on out-of-distribution (OOD) data**. Even recent universal models like Unicorn-U [52], despite being trained for versatility, fall short due to two key issues: they rely on non-unified architectures, and their performance still

[*] Contributed equally to this work. [†]Corresponding author.

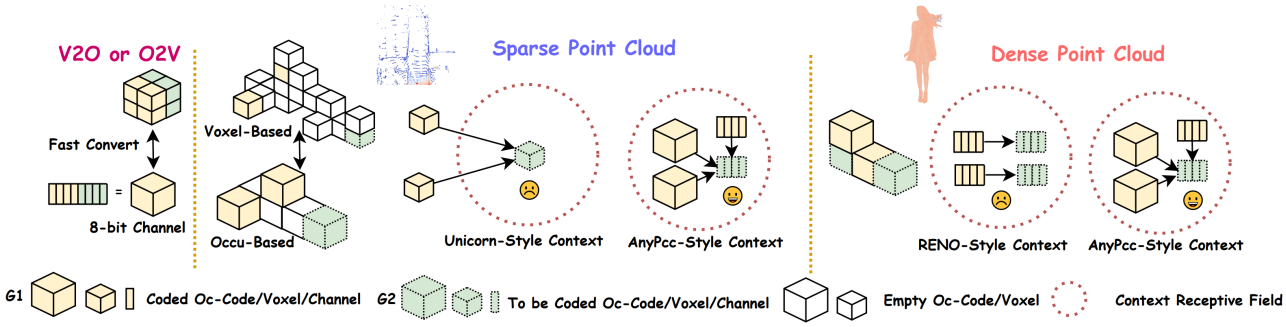

Figure 2. The superiority of our UCM in capturing contextual information across diverse point cloud types (e.g., dense and sparse).

collapses under extreme OOD conditions. We address these limitations sequentially, first by tackling context modeling.

Effective context modeling is crucial for efficient point cloud compression. While spatial-prior methods like Unicorn [52] leverage strong contextual information at the voxel scale, they become unreliable in sparse scenarios. RENO [66] re-frames the problem as a channel-wise probability prediction task on occupancy codes. Although their methodologies differ, the spatial context modeled by Unicorn and the channel-wise context by RENO are fundamentally two inter-convertible representations of the same local geometry. Operating at the coarser-grained occupancy code scale ensures sparsity-robustness and achieves a larger effective receptive field. However, by exclusively grouping fine-grained priors, both methods fail to leverage the valuable structural information available from coarser-grained contexts, as illustrated in Figure 2. We resolve this with our Universal Context Model (UCM), which synergistically integrates channel grouping for **fine-grained priors** with spatial grouping for **coarse-grained structural priors**. This unified approach allows the UCM to capture a richer, more complete set of contextual dependencies, enabling robust compression across the entire density spectrum.

Although our UCM achieves favorable compression results via large-scale pre-training, its performance still degrades on ***extreme*** OOD samples. Meanwhile, while some works have explored Implicit Neural Representations (INRs) for generalization [6, 11, 23, 43], their requirement to train a unique network from scratch for every instance results in impractically long encoding times. Therefore, to transcend the slow encoding time of INRs and the generalization constraints of a fixed pre-trained model, we propose **AnyPcc**. Our key idea is to combine a versatile and powerful pre-trained model with fast, instance-specific adaptation. To this end, our Instance-Adaptive Fine-Tuning (IAFT) strategy updates only a minimal subset of the network's parameters (the final linear layers), leaving the vast majority untouched. This lightweight procedure enables AnyPcc to converge to an effective instance-specific model within seconds which facilitated by the robust priors of our UCM. This approach transcends the slow encoding of traditional INRs and the generalization limits of a fixed pre-trained model, achieving an outstanding balance between compression performance, generalization, and practical efficiency. To validate this, we curated a comprehensive training corpus and established a demanding evaluation benchmark that extends far beyond standard test sets, providing a rigorous measure of its real-world viability, as shown in Figure 1. Our contributions are summarized as follows:

- We introduce **AnyPcc**, a universal compression framework designed to resolve the critical trade-off between efficiency and generalization. It is the first method to achieve high compression and robust performance across diverse point cloud types using **a single, unified model**.
- Our **Universal Context Model (UCM)** resolves a key limitation of prior works by being the first to synergistically integrate fine-grained channel priors with coarse-grained spatial priors, achieving robust context modeling across the entire density spectrum.
- We pioneer an **Instance-Adaptive Fine-Tuning (IAFT)** strategy that resolves the trade-off between explicit and implicit compression. It rapidly fine-tunes a small subset of a pre-trained model's weights per-instance, yielding a highly efficient compression model in seconds.
- Extensive experiments on a benchmark of **15 diverse datasets** demonstrate that AnyPcc not only outperforms the latest G-PCC v23 standard but also sets a new state-of-the-art among learning-based methods, establishing a new paradigm for universal point cloud compression.

2. Related Work

2.1. Point Cloud Geometry Compression

Category-Specific Methods. In the domain of dense object point cloud compression [15, 16, 37, 41, 61–63, 65], several methods leverage sparse tensor representations. For instance, PCGCv2 [50], SparsePCGC [51], UniPCGC [54] and methods [3, 13, 30, 40, 68, 70, 73] employ sparse convolutions to exploit local correlations for effective com-

pression. Concurrently, approaches like VoxelDNN [38] and SparseVoxelDNN [36] achieve higher compression efficiency through autoregressive codecs for feature extraction. For LiDAR point clouds compression [1, 7, 8, 25, 33, 42, 46, 60, 67], seminal works such as MuSCLE [4], Octsqueeze [22], OctAttention [14], EHEM [45] and TopNet [58] utilize octree-based [44] representations, employing attention mechanisms to predict the occupancy codes of parent nodes efficiently. Other methods, including Unicorn [52] and RENO [66], also adapt the sparse tensor paradigm to LiDAR data. More recently, with the rise of 3D Gaussian Splatting [26, 28, 32] (3DGS), GausPcgc [55] is introduced, which enables efficient compression of 3DGS coordinates. Nevertheless, a unifying drawback common to all of these approaches is their architectural over-specialization for particular point cloud attributes, which in turn severely compromises their generalizability.

The Generalization Problem. Despite recent efforts toward versatility from frameworks like Unicorn [52] and standardization bodies (MPEG [27], AVS, JPEG [18] AI-PCC), a critical generalization problem persists. Unicorn-U [52] serves as a prime example, exhibiting two critical flaws: it relies on a non-unified, hybrid architecture (e.g., attention and convolution) that compromises practical utility, and its performance collapses on out-of-distribution (OOD) data. This dependency on curated training data is untenable for the vast and varied landscape of real-world point clouds. Dedicated training data is often unavailable for many critical types, such as medical scans, 3D Gaussian Splats, or point clouds generated by networks like Dust3R [57] and VGGT [53]. Consequently, the compression efficiency of existing methods degrades drastically on such OOD data. This performance collapse reveals a critical, yet largely unaddressed, challenge in the current research landscape: ***achieving true generalization capability in point cloud compression networks.***

2.2. Implicit Compression

Implicit neural representations (INRs) offer strong generalization by overfitting a coordinate-based network (e.g., an MLP) to a single data instance, using the optimized weights as the compressed representation [6, 11, 12, 35, 43, 49, 64, 71, 72]. However, this per-instance optimization is prohibitively slow, as it requires full network training, and the process is often lossy.

Some works have sought to mitigate these issues. For instance, while LINR-PCGC [23] achieves lossless compression by overfitting a base model, it requires full network tuning and is limited to multi-frame scenarios due to the impractical storage overhead of per-frame weights. In image and video compression, recent works [34, 39, 48] have employed parameter-efficient fine-tuning (PEFT) techniques [21, 56] to adaptively fine-tune pre-trained models. This hybrid paradigm, which synergizes the robust priors of a pre-trained model with the flexibility of implicit, per-instance adaptation, remains unexplored in point cloud compression. The highly irregular structure of point clouds further necessitates a dedicated investigation into designing an effective explicit-implicit model tailored for this domain.

3. Method

3.1. Overview

Our proposed framework, AnyPcc, achieves universal point cloud compression through three core components. First, we introduce a **Universal Context Model (UCM)**, pre-trained on diverse data to adaptively handle point clouds of varying densities. Second, we employ an **Instance-Adaptive Fine-tuning** strategy to efficiently compress out-of-distribution (OOD) samples. Finally, we demonstrate how our framework can be seamlessly extended from lossless to lossy compression via a probability thresholding mechanism, **creating a single, unified solution**. These components are detailed in the following subsections.

3.2. Universal Context Model

Design Insight. The task of point cloud geometry compression can be viewed as predicting occupancy probabilities, where a more accurate model yields a shorter bitstream by arithmetic coding. While prior work has leveraged priors effectively, it has focused exclusively on fine-grained information. While spatial context models [51, 54] focus on the voxel scale and channel-wise context models [66] on the occupancy code scale, they share a fundamental limitation. Both overlook the coarse-grained structural context available from the spatial relationships between the occupancy codes themselves. Our UCM resolves this by introducing a synergistic partitioning scheme that is the first to simultaneously model fine-grained dependencies via channel grouping and coarse-grained context via spatial grouping. This unified approach creates a model robust across the entire density spectrum, as shown in Figure 2 and 3.

Our spatio-channel design is grounded in two theoretical principles, which we prove in *Appx.* 6. We first establish that modeling on 8-bit occupancy code channels is information-theoretically equivalent to modeling on the corresponding $2 \times 2 \times 2$ voxel block (Theorem 1). We then prove this provides a significant receptive field advantage crucial for sparse data (Theorem 2). Leveraging these principles, the UCM employs a hierarchical framework over a multi-scale occupancy code representation. The prediction proceeds scale-by-scale, from coarse to fine, where the context at each scale is drawn from three sources: the occupancy codes of the parent scale, fine-grained channel-wise priors, and coarse-grained spatial priors. Critically, unlike in image models [19, 20, 24], our context grouping operates

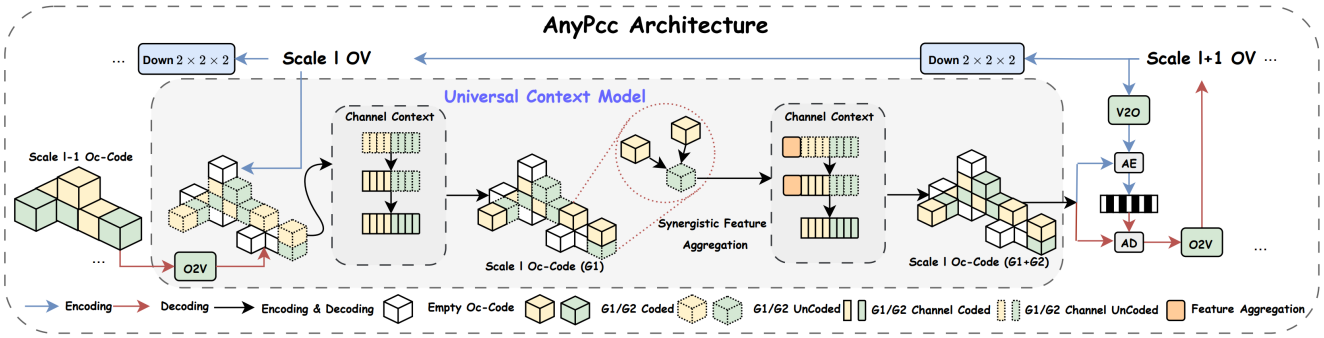

Figure 3. llustration of the proposed AnyPcc framework. AnyPcc formulates the compression task as a sequential prediction of multi-scale occupancy codes. It develops a **Universal Context Model (UCM)** that incorporates strong inductive biases from spatio-channel partitioning, enabling robust generalization across diverse data sources.

directly on geometric occupancy codes instead of a latent space, creating a true coarse-to-fine geometric partitioning. This design ensures robust prediction, as detailed in Figure 4. (More in *Appx.* 7.1)

Notation. V2O/O2V denote the Voxel-to-Occupancy-Code conversion and vice-versa (Fig. 2). Oc-Code is an occupancy code and OV is an occupied voxel. The Arithmetic Encoder (AE) compresses Oc-Codes to a bitstream using predicted probabilities and the Arithmetic Decoder (AD) reconstructs them from the bitstream.

Hierarchical Context Propagation. Our model adopts a coarse-to-fine hierarchical structure. At each scale l, the geometric information is captured by a set of occupied voxel coordinates $\mathcal{C}^{(l)}$ and their associated 8-bit occupancy codes $\{o_i^{(l)}\}$. The context propagation mechanism, involves two main stages. First, a coarse-context encoding network, Ψ_{prior}, processes the occupancy codes and coordinates from scale l to generate a powerful latent representation $\mathbf{Z}^{(l)}$. This network implicitly handles the conversion from discrete codes to continuous features and aggregates local contextual information:

$$\mathbf{Z}^{(l)} = \Psi_{\text{prior}}(\{o_i^{(l)}\}, \mathcal{C}^{(l)}). \tag{1}$$

Second, this latent representation is propagated to the finer resolution scale $l+1$. The features are upsampled and then refined by a target-processing network Ψ_{target} to produce the final predictive context $\{\mathbf{c}_j^{(l+1)}\}$ at the finer scale:

$$\{\mathbf{c}_i^{(l+1)}\} = \Psi_{\text{target}}(\text{Upsample}(\mathbf{Z}^{(l)})), \tag{2}$$

where Ψ_{target} and Ψ_{prior} are shown in Figure 4. The resulting context field $\{\mathbf{c}_i^{(l+1)}\}$ provides the foundation for our detailed spatio-channel prediction task.

Spatio-Channel Context Factorization. After aggregating information from a coarser scale l, the compression task at the current scale $l+1$ is to predict its occupancy codes. To efficiently model this, we decompose the joint probability distribution of the occupancy codes by factorizing their context along both spatial and channel-wise dimensions. This dual factorization dramatically reduces predictive complexity and improves coding efficiency.

Spatial Context Partitioning. To model the coarse-grained structural priors, we first spatially partition the target occupancy codes at scale $l+1$ into two disjoint sets using a 3D checkerboard pattern, as shown in Figure 3. An occupancy code at location (x, y, z) is assigned to group $\mathcal{G}_1$ (yellow cells) if the coordinate sum $(x+y+z)$ is even, and to group $\mathcal{G}_2$ (green cells) if it is odd:

$$\mathcal{G}_1 = \{i \in \mathcal{V}^{(l+1)} \mid (x_i + y_i + z_i) \pmod 2 = 0\}, \tag{3}$$

$$\mathcal{G}_2 = \{i \in \mathcal{V}^{(l+1)} \mid (x_i + y_i + z_i) \pmod 2 = 1\}. \tag{4}$$

This partitioning strategy transforms the prediction task into a two-step auto-regressive process. The joint probability of all $\{o_i\}$ at the current scale $l+1$ is factorized as:

$$P(\{o_i\}) = \prod_{i \in \mathcal{G}_1} P(o_i | \mathbf{c}_i) \cdot \prod_{j \in \mathcal{G}_2} P(o_j | \mathbf{c}_j, \{o_k\}_{k \in \mathcal{N}(j) \cap \mathcal{G}_1}), \tag{5}$$

where $\mathbf{c}_i$ and $\mathbf{c}_j$ are the context vectors for locations $i \in \mathcal{G}_1$ and $j \in \mathcal{G}_2$ respectively, drawn from the context field $\{\mathbf{c}_k^{(l+1)}\}$ propagated from the coarser scale l. $\mathcal{N}(j)$ is the k^3 neighborhood of location j. This spatial factorization establishes a powerful dependency model on the coarse-grained context, allowing the network to leverage information from immediately adjacent, already-decoded occupancy codes for more accurate probability estimation.

Channel Context Partitioning. Complementary to the coarse-grained spatial factorization, we introduce a channel-wise partitioning to model the fine-grained context within each 8-bit occupancy code $o \in \{0, \ldots, 255\}$. The code is decomposed into two 4-bit sub-symbols: a least sig-

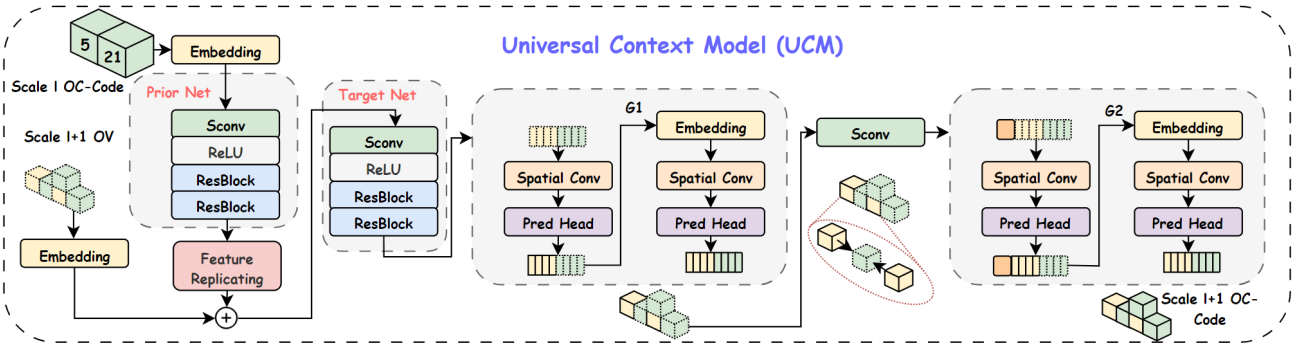

Figure 4. An illustration of our recursive, parameter-shared Universal Context Model (UCM) operating at scale $l+1$. At each scale, the UCM processes the input voxels to predict the probability distribution of the corresponding occupancy codes. The same model architecture and parameters are recursively applied to all other scales, ensuring a consistent prediction process throughout the hierarchy.

nificant part o_0 and a most significant part o_1.

$$o_0 = o \pmod{16}, \quad o_1 = \lfloor o/16 \rfloor. \tag{6}$$

This channel-wise partitioning enables a two-stage prediction cascade. For any location i, the probability of its occupancy code o_i is factorized as:

$$P(o_i|\cdot) = P(o_{i,0}|\cdot) \cdot P(o_{i,1}|o_{i,0}, \cdot), \tag{7}$$

where $(\cdot)$ represents the full conditioning context available at location i. For instance, for a location $i \in \mathcal{G}_1$, this context is simply $\mathbf{c}_i$. The sub-symbol probabilities are modeled sequentially via a prediction module composed of a Spatial Convolution $\mathcal{S}$ and a Prediction Head $\mathcal{H}$. For the second sub-symbol, the context is updated to $\mathbf{c}'_i = \mathbf{c}_i + \text{Emb}(o_{i,0})$. The probabilities are then given by:

$$P(o_{i,0}|\mathbf{c}_i) \propto \exp\left(\mathcal{H}_{\text{g1},0}(\mathcal{S}_{\text{g1},0}(\mathbf{c}_i))\right), \tag{8}$$

$$P(o_{i,1}|o_{i,0}, \mathbf{c}_i) \propto \exp\left(\mathcal{H}_{\text{g1},1}(\mathcal{S}_{\text{g1},1}(\mathbf{c}'_i))\right). \tag{9}$$

This cascading mechanism improves coding efficiency with refining the context for the second sub-symbol with the first. The process is analogous for $\mathcal{G}_2$, which additionally incorporates spatial context from decoded neighbors in $\mathcal{G}_1$.

Synergistic Context Feature Aggregation. The synergy between the spatial partitions is realized when predicting the second group $\mathcal{G}_2$. The initial context for a location $j \in \mathcal{G}_2$ is explicitly enhanced by aggregating feature from its decoded neighbors in $\mathcal{G}_1$, as highlighted in Figure 3. This process involves three steps:

First, we create a sparse, augmented context field $\hat{\mathbf{C}}$ populated only at locations corresponding to the already-decoded group $\mathcal{G}_1$. For each location $k \in \mathcal{G}_1$, we augment its initial context $\mathbf{c}_k$ with an embedding of its decoded occupancy code o_k:

$$\hat{\mathbf{c}}_k = \mathbf{c}_k + \text{Emb}_{\text{prior}}(o_k), \quad \forall k \in \mathcal{G}_1. \tag{10}$$

Second, this local information is gathered for each location $j \in \mathcal{G}_2$ using a sparse convolution (Sconv). The aggregated context $\mathbf{c}_{\text{agg},j}$ is computed by applying a weighted sum over the augmented contexts of its neighbors in $\mathcal{G}_1$:

$$\mathbf{c}_{\text{agg},j} = \left[\text{Sconv}(\hat{\mathbf{C}})\right]_j = \sum_{k \in \mathcal{N}(j) \cap \mathcal{G}_1} \mathbf{W}_{\text{sconv}}(k-j) \cdot \hat{\mathbf{c}}_k. \tag{11}$$

Finally, the enhanced context $\mathbf{c}''_j$ for location j is formed by fusing its original context $\mathbf{c}_j$ with the aggregated neighbor context $\mathbf{c}_{\text{agg},j}$ via a fusion network Φ_{fuse}:

$$\mathbf{c}''_j = \Phi_{\text{fuse}}([\mathbf{c}_j, \mathbf{c}_{\text{agg},j}]), \tag{12}$$

where $[\cdot,\cdot]$ denotes concatenation. This process allows the model to directly leverage the decoded geometric structure from $\mathcal{G}_1$ to resolve ambiguities and improve predictions for the subsequent group $\mathcal{G}_2$. After all partitions at scale $l+1$ have been processed, the entire procedure is repeated for the next finer scale $l+2$.

3.3. Instance-Adaptive Fine-Tuning

While the pre-trained UCM provides a strong general prior, its performance will degrade when encountering OOD instances. To bridge this generalization gap, we introduce Instance-Adaptive Fine-Tuning (IAFT), a novel strategy that synergizes the strengths of a general pre-trained model with the instance-adaptive finetuning.

Parameter-Efficient Adaptation. The core idea of IAFT is to specialize the UCM to a new instance through parameter-efficient fine-Tuning. To achieve this, we partition the UCM's parameters Θ into a large, frozen backbone Θ_{frozen} and a small, tunable set Θ_{tune}. The frozen set Θ_{frozen} comprises the vast majority of the network, including all feature extraction modules and the spatial convolution operators $\mathcal{S}$. These components provide a robust general prior. Conversely, the tunable set Θ_{tune} consists exclusively of the parameters of the lightweight prediction heads $\mathcal{H}$, which

are responsible for mapping the final features to occupancy probabilities.

On-the-Fly Optimization. For each new point cloud instance, we perform a rapid, on-the-fly optimization to find a specialized set of weights Θ^*_{tune} that minimizes an instance-specific loss function. This process is formulated as:

$$\Theta^*_{\text{tune}} = \arg\min_{\Theta_{\text{tune}}} \mathcal{L}_{\text{tune}}, \tag{13}$$

where the loss $\mathcal{L}_{\text{tune}}$ is the negative log-likelihood of the instance's occupancy codes $\{o_i\}$, which directly models the bitrate, regularized by an L1 term to promote sparsity:

$$\mathcal{L}_{\text{tune}} = -\sum_i \log P(o_i|\text{context}_i;\Theta) + \lambda_{\text{L1}}\|\Theta_{\text{tune}}\|_1. \tag{14}$$

To make this adaptation highly efficient, we employ a caching mechanism. We first perform a single forward pass through the frozen backbone Θ_{frozen} to compute and cache the inputs to the prediction heads. The subsequent optimization iterations then operate solely on these cached features to rapidly update the small Θ_{tune} set. Thanks to the powerful priors from the backbone, this process converges in just a few hundred gradient steps (mere seconds) and yielding a highly accurate, instance-specific predict model.

Encoding and Decoding. The final compressed bitstream is composed of two components:

- **Model Component ($\mathcal{B}_{\text{weights}}$):** We employ uniform scalar quantization with a step size $\Delta = 2^{-0.5\times16}$ to the weights Θ^*_{tune} , followed by `DeepCABAC` coding [59].
- **Geometry Component ($\mathcal{B}_{\text{geom}}$):** The occupancy codes $\{o_i\}$ are losslessly compressed by an arithmetic coder guided by the probability distributions from the now-specialized UCM.

The total bitrate is the sum of the lengths of these two components. The decoding process is symmetric: the decoder first reads $\mathcal{B}_{\text{weights}}$ to reconstruct the specialized prediction heads, updates its local copy of the UCM, and then uses this instance-specific model to entropy-decode the geometry from $\mathcal{B}_{\text{geom}}$. By transmitting a small model update, the significant bitrate reduction in encoding the geometry far outweighs the overhead of sending the weights, leading to substantial gains in coding efficiency and generalization.

3.4. Unified Lossless and Lossy Compression

Although our UCM framework is primarily designed for lossless compression, it can be seamlessly extended to lossy scenarios. For sparse LiDAR point clouds, a straightforward lossy approach is to simply omit the entropy coding of the n finest scales. However, this method would cause severe geometric degradation for dense point clouds. Therefore, We employ a more sophisticated strategy for dense data: the encoder transmits only the ground-truth point count (k) for a given scale. The decoder then reconstructs the geometry by identifying the k most probable occupied locations and their corresponding occupancy codes based on the model's predictions. Please refer to *Appx.* 7.2 for detailed model specifications and *Appx.* 9.4 for an analysis of practical deployment trade-offs.

4. Experimental Results

4.1. Experiment Setup

Dataset. To cultivate a powerful and generalizable Universal Context Model (UCM), we curate a comprehensive training corpus by merging multiple datasets. This corpus includes a wide array of point clouds recommended by the MPEG AI-PCC and AVS AI-PCC working groups, such as KITTI [17], Ford [2], 8iVFB [10], MVUB [31], and ScanNet [9]. To further diversify the training data and enhance model robustness, we also incorporate the GausPcc-1K [29, 55] and Thuman [69] datasets. To truly challenge our proposed framework and assess its real-world viability, our evaluation protocol goes far beyond standard benchmarks. In addition to all mainstream datasets, we deliberately introduce point clouds from modern reconstruction techniques like VGGT methods and 3D Gaussian Splatting. Furthermore, we synthesize three challenging datasets to simulate common data imperfections: NS (added noise), RS (point dropout), and CS (non-rigid shape deformations). This comprehensive and demanding benchmark provides a more realistic measure of a model's stability and generalization capabilities (see *Appx.* 8).

Implementation. Our framework is built with PyTorch and TorchSparse [47], and all experiments are conducted on a single NVIDIA RTX 3090 GPU. We propose and evaluate two distinct versions of our model:

- **Ours:** This version aligns with standard practice by training a **dedicated model** for each dataset category.
- **Ours-U:** This is a single, **unified model** trained on a large-scale mixed dataset. The same set of weights is applied to all test sets, greatly improving its practical utility.

For both versions, we apply our UCM on E samples, and augment it with IAFT (200 iters) for M and H samples.

Baseline and Metrics. We benchmark our method against several state-of-the-art (SOTA) open-source solutions, including RENO [66], SparsePCGC [51], OctAttention [14], and TopNet [58], as well as the latest traditional codec, GPCC v23. Since the code for Unicorn [52] is unavailable, we report its performance from the original paper under aligned experimental settings. All models are evaluated under identical training and testing conditions for a fair comparison. For lossless compression, we report the efficiency in bits per point (bpp) and the Compression Ratio Gain (CR-Gain). The CR-Gain is calculated relative to an

Table 1. Performance comparison on the AnyPcc Benchmark. The table presents the compression performance of AnyPcc against six methods across 15 diverse datasets, with the best and second-best results highlighted in red and yellow cells.

Dataset	Cond†	OOD	RENO	SparsePCGC	Unicorn*	OctAttention	TopNet	GPCC	Ours	Ours-U
8iVFB		✗	0.70	0.57	0.57	0.68	0.59	0.76	0.54	0.57
MVUB		✗	1.00	0.69	0.69	0.76	0.69	0.94	0.67	0.75
Owlii		✗	0.59	0.48	0.48	0.62	0.56	0.59	0.47	0.47
Thuman	E	✗	1.64	1.70	1.70	2.31	2.20	2.00	1.58	1.64
ScanNet		✗	2.15	1.86	1.86	2.13	2.03	2.03	1.83	1.88
KITTI		✗	7.06	6.80	6.50	7.21	6.85	8.19	6.18	6.45
Ford		✗	9.38	9.77	8.44	9.10	8.54	10.32	8.40	8.57
Dense		✗	5.81	6.37	5.48	6.55	6.38	5.32	5.27	5.55
Sparse		✗	9.64	9.98	9.42	10.40	10.02	9.35	9.11	9.26
GS	M	✗	13.89	15.82	/	11.31	10.95	14.46	11.65	11.74
VGGT		✓	8.24	7.84	/	8.22	7.83	7.33	7.30	7.06
S3DIS		✓	13.06	11.88	/	11.52	10.84	10.66	10.93	10.79
RS		✓	4.02	3.88	/	4.05	3.92	3.72	3.68	3.50
NS	H	✓	4.96	6.54	/	4.89	4.84	4.85	4.69	4.67
CS		✓	3.94	4.94	/	3.40	3.21	3.23	3.18	3.08
CR Gain over GPCC ↓			2.96%	2.07%	/	1.32%	-4.04%	0.00%	-11.93%	-10.75%
Enc/Dec Time (s) ↓			0.22/0.23	2.6/2.2	/	7.7/1324	8.7/1740	3.8/2.7	2.84/0.46	
Total Parameters (M) ↓			9.03	26.43	/	29.61	23.59	/	68.39	9.77

† Cond represents the test difficulty of the testsets, and we divide the test set into easy (E), medium (M), and hard (H).
* The results for Unicorn are cited directly from the original publication as its implementation is not open-source.

anchor codec as $(\text{bpp}_{\text{method}} - \text{bpp}_{\text{anchor}})/\text{bpp}_{\text{anchor}} \times 100\%$, where a more negative value indicates greater bitrate savings. For lossy compression, we assess the rate-distortion performance using bpp for the rate and Peak Signal-to-Noise Ratio (PSNR) for the distortion.

4.2. Lossless Compression

Results. As benchmarked in Table 1 across 15 datasets (including 5 for Out-of-Distribution generalization), our methods demonstrate clear superiority. Our models, **Ours** and **Ours-U**, achieve SOTA on 13 datasets and deliver substantial CR-Gain of **11.93%** and **10.75%** over the GPCC v23 anchor. In stark contrast, most baselines fail to match this anchor: RENO, SparsePCGC, and OctAttention show performance degradations of 2.96%, 2.07%, and 1.32% relative to the anchor, while TopNet is the only competing method to provide a positive gain, at 4.04%. A key trade-off emerges: our specialized model excels on in-distribution data, but our universal model (**Ours-U**) shows superior generalization across all OOD datasets, affirming its practical value. For OOD evaluation, models trained on KITTI are used. Moreover, we provide both the results for RENO-U and SparsePCGC-U with unified data training and the more evaluations on dense point clouds in *Appx.* 9.1.

Model Parameters. Our primary method (Ours), like all baselines, requires training seven distinct models for the benchmark, as shown in Table 1. In contrast, our universal model (Ours-U) utilizes a single set of weights for all datasets. This not only highlights its generalization but also drastically reduces storage and deployment overhead, marking a significant step towards a practical solution.

Codec Time. Our decoding time is comparable to RENO, the fastest baseline, as shown in Table 1. While our default encoding takes 2.84 seconds, it is highly flexible. By adjusting fine-tuning iterations, the encoding time can be controlled within a 0.44s to 2.84s range, offering a trade-off between speed and compression efficiency. A detailed analysis is in the *Appx.* 9.3.

4.3. Lossy Compression

The lossy compression results are shown in Figure 5. LiDAR-based methods are not shown for the dense human-body and ScanNet datasets due to their suboptimal performance. The results confirm that AnyPcc performs robustly across all datasets. Crucially, the strong performance of our unified model, Ours-U, demonstrates that AnyPcc can serve as a single-model solution for both efficient lossless and lossy point cloud compression. For an analysis of model parameters and codec times in the lossy compression, please refer to the *Appx.* 9.2.

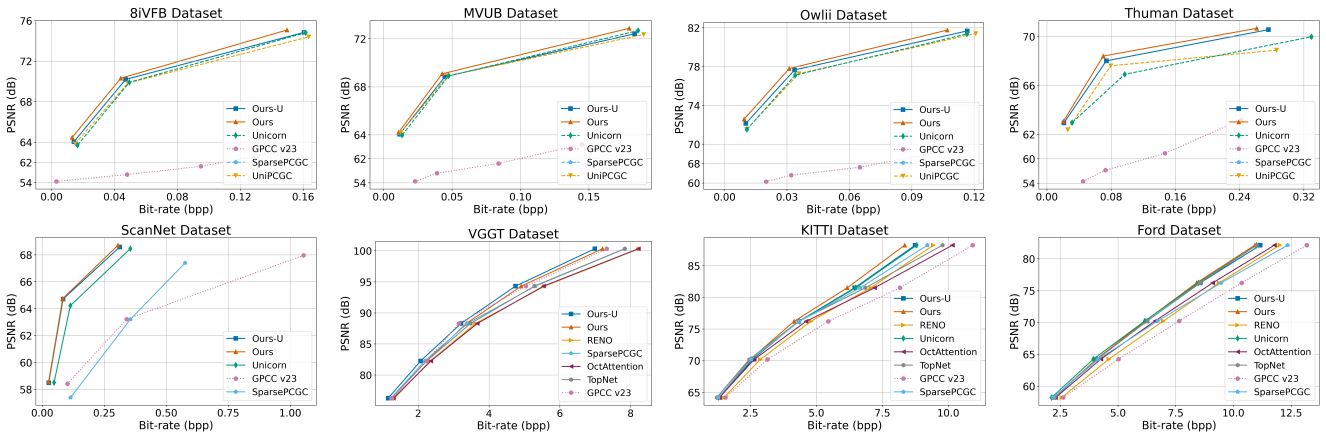

Figure 5. Performance comparison using rate-distortion curves. Comparisons on the human-body and ScanNet datasets exclude RENO, OctAttention, and TopNet, as they are specifically designed for lossy LiDAR compression and thus incompatible.

Table 2. Effectiveness analysis of key modules in UCM.

	SC	SG	CG	CR-Gain	Params (M)
Baseline	✗	✗	✗	0.00%	5.15
Abla1	✗	✓	✗	-6.56%	5.68
Abla2	✗	✗	✓	0.13%	5.15
Abla3	✓	✓	✗	-7.74%	9.78
Abla4	✓	✗	✓	-5.33%	7.19
Abla5	✗	✓	✓	-6.50%	5.67
Ours	✓	✓	✓	**-9.88**%	9.77

Table 3. Ablation study of channel count and IAFT module, including Bpp composition on the GS dataset.

Metrics	UCM	UCM+IAFT
Overall Performance (32/64/128 channels)		
Bpp	5.42/5.32/5.25	5.14/5.04/4.99
Enc Time (s)	0.31/0.44/1.13	11.82/12.11/13.08
Dec Time (s)	0.32/0.46/1.14	0.32/0.46/1.14
Model Size (M)	2.45/9.77/39.01	
Bpp Composition on GS dataset		
Entropy Coding Bpp	13.307	11.424
Net Size Bpp	0	0.319

4.4. Ablation Studies

Key Modules. Table 2 details our ablation study on Spatial Convolution (SC), Spatial Grouping (SG), and Channel Grouping (CG). Combining all three yields the optimal performance. Notably, while SC and SG individually enhance results, using CG alone (as in RENO) causes degradation. This indicates that CG's fine-grained priors strictly depend on the coarse-grained spatial priors from SC and SG.

Channel Count. Table 3 shows the effect of varying the channel count (32 / 64 / 128) on performance and complexity. While higher channel counts lead to better performance, the model with 128 channels proves to be too computationally expensive. We therefore choose C=64 for our model, as it offers the best trade-off between performance and speed.

IAFT Module. We also conduct an ablation study on the Instance-Adaptive Fine-Tuning (IAFT) module, with the results presented in Table 3. The table shows that fine-tuning for 800 iterations yields a performance gain of approximately 5%, at the cost of an additional 10 seconds in encoding time. This allows for dynamic control over the complexity-performance trade-off. As shown in Table 3, IAFT reduces the entropy-coded bpp by 1.883 on the GS dataset, at the cost of 0.319 bpp for the model. This confirms that for samples with irregular distributions, the bitrate savings from entropy coding substantially outweigh the network overhead.

More Studies. Further evaluations (e.g., on quantization precision, fine-tuning iterations, training data ratio, context model) and visual comparisons are detailed in *Appx.* 9.3.

5. Conclusion

This paper introduced **AnyPcc**, a universal framework for geometry point cloud compression designed to efficiently compress point clouds of any type. Powered by a robust generic context model and an instance-adaptive fine-tuning strategy, our approach uniquely combines the advantages of both explicit and implicit compression methods. This allows for effective compression of both in-distribution (ID) and out-of-distribution (OOD) samples. Experimental results on a comprehensive benchmark of 15 datasets show that AnyPcc achieves SOTA compression performance.

Acknowledgment. This work was supported by The Major Key Project of PCL (PCL2024A02), Natural Science Foundation of China (62271013), Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (2024B1212010006), Guangdong Province Pearl River Talent Program (2021QN020708), Guangdong Basic and Applied Basic Research Foundation (2024A1515010155), Shenzhen Science and Technology Program (JCYJ20240813160202004, JCYJ20230807120808017, SYSPG20241211173440004). (*Corresponding author: Wei Gao*)

References

[1] Rashid Abbasi, Ali Kashif Bashir, Hasan J Alyamani, Farhan Amin, Jaehyeok Doh, and Jianwen Chen. Lidar point cloud compression, processing and learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 24(1):962–979, 2022. 3

[2] Siddharth Agarwal, Ankit Vora, Gaurav Pandey, Wayne Williams, Helen Kourous, and James McBride. Ford multi-av seasonal dataset. *The International Journal of Robotics Research*, 39(12):1367–1376, 2020. 6

[3] Anique Akhtar, Zhu Li, and Geert Van der Auwera. Inter-frame compression for dynamic point cloud geometry coding. *IEEE Transactions on Image Processing*, 33:584–594, 2024. 2

[4] Sourav Biswas, Jerry Liu, Kelvin Wong, Shenlong Wang, and Raquel Urtasun. Muscle: Multi sweep compression of lidar using deep entropy models. *Advances in Neural Information Processing Systems*, 33:22170–22181, 2020. 3

[5] Chao Cao, Marius Preda, Vladyslav Zakharchenko, Euee S Jang, and Titus Zaharia. Compression of sparse and dense dynamic point clouds—methods and standards. *Proceedings of the IEEE*, 109(9):1537–1558, 2021. 1

[6] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021. 2, 3

[7] Mingyue Cui, Junhua Long, Mingjian Feng, Boyang Li, and Huang Kai. Octformer: Efficient octree-based transformer for point cloud compression with local enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 470–478, 2023. 3

[8] Mingyue Cui, Yuyang Zhong, Mingjian Feng, Junhua Long, Yehua Ling, Jiahao Xu, and Kai Huang. Gaem: Graph-driven attention-based entropy model for lidar point cloud compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 3

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 6

[10] Eugene d'Eon, Harrison Bob, Taos Myers, and Philip A. Chou. 8i voxelized full bodies - a voxelized point cloud dataset. In *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006*, 2017. 6

[11] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021. 2, 3

[12] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goliński, Yee Whye Teh, and Arnaud Doucet. Coin++: Neural compression across modalities. *arXiv preprint arXiv:2201.12904*, 2022. 3

[13] Tingyu Fan, Linyao Gao, Yiling Xu, Zhu Li, and Dong Wang. D-dpcc: Deep dynamic point cloud compression via 3d motion prediction. *arXiv preprint arXiv:2205.01135*, 2022. 2

[14] Chunyang Fu, Ge Li, Rui Song, Wei Gao, and Shan Liu. Octattention: Octree-based large-scale contexts model for point cloud compression. In *Proceedings of the AAAI conference on artificial intelligence*, pages 625–633, 2022. 3, 6

[15] Wei Gao, Liang Xie, Songlin Fan, Ge Li, Shan Liu, and Wen Gao. Deep learning-based point cloud compression: An in-depth survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2

[16] Wenxu Gao, Liang Xie, Kangli Wang, Jingxuan Su, Changhao Peng, and Wei Gao. Dpcset: A large-scale dynamic point cloud dataset for compression and perception. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 13384–13390, 2025. 2

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013. 6

[18] André FR Guarda, Nuno MM Rodrigues, and Fernando Pereira. The jpeg pleno learning-based point cloud coding standard: Serving man and machine. *IEEE Access*, 2025. 3

[19] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14771–14780, 2021. 3

[20] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5718–5727, 2022. 3

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3

[22] Lila Huang, Shenlong Wang, Kelvin Wong, Jerry Liu, and Raquel Urtasun. Octsqueeze: Octree-structured entropy model for lidar compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1313–1323, 2020. 3

[23] Wenjie Huang, Qi Yang, Shuting Xia, He Huang, Zhu Li, and Yiling Xu. Linr-pcgc: Lossless implicit neural representations for point cloud geometry compression. *arXiv preprint arXiv:2507.15686*, 2025. 2, 3

[24] Wei Jiang, Jiayu Yang, Yongqi Zhai, Peirong Ning, Feng Gao, and Ronggang Wang. Mlic: Multi-reference entropy model for learned image compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7618–7627, 2023. 3

[25] Yiqi Jin, Ziyu Zhu, Tongda Xu, Yuhuan Lin, and Yan Wang. Ecm-opcc: Efficient context model for octree-based point cloud compression. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7985–7989. IEEE, 2024. 3

[26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3

[27] Ge Li, Wei Gao, and Wen Gao. Mpeg ai-based 3d graphics coding standard. In *Point Cloud Compression: Technologies and Standardization*, pages 219–241. Springer, 2024. 3

[28] Junhong Lin, Kangli Wang, Shunzhou Wang, Songlin Fan, Ge Li, and Wei Gao. Vgd: Visual geometry gaussian splatting for feed-forward surround-view driving reconstruction. *arXiv preprint arXiv:2510.19578*, 2025. 3

[29] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 6

[30] Bojun Liu, Yangzhi Ma, Ao Luo, Li Li, and Dong Liu. Voxel-based point cloud geometry compression with space-to-channel context. *arXiv preprint arXiv:2503.18283*, 2025. 2

[31] Charles Loop, Qin Cai, Sergio Orts Escolano, and Philip 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 M*, 72012, 2021. 6

[32] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 3

[33] Ao Luo, Linxin Song, Keisuke Nonaka, Kyohei Unno, Heming Sun, Masayuki Goto, and Jiro Katto. Scp: spherical-coordinate-based learned point cloud compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3954–3962, 2024. 3

[34] Yue Lv, Jinxi Xiang, Jun Zhang, Wenming Yang, Xiao Han, and Wei Yang. Dynamic low-rank instance adaptation for universal neural image compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 632–642, 2023. 3

[35] Yuzhen Mao, Zhun Deng, Huaxiu Yao, Ting Ye, Kenji Kawaguchi, and James Zou. Last-layer fairness fine-tuning is simple and effective for neural networks. *arXiv preprint arXiv:2304.03935*, 2023. 3

[36] Dat Thanh Nguyen and André Kaup. Learning-based lossless point cloud geometry coding using sparse tensors. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2341–2345. IEEE, 2022. 3

[37] Dat Thanh Nguyen, Maurice Quach, Giuseppe Valenzise, and Pierre Duhamel. Multiscale deep context modeling for lossless point cloud geometry compression. *arXiv preprint arXiv:2104.09859*, 2021. 2

[38] Dat Thanh Nguyen, Maurice Quach, Giuseppe Valenzise, and Pierre Duhamel. Learning-based lossless compression of 3d point cloud geometry. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4220–4224. IEEE, 2021. 3

[39] Seungjun Oh, Hyunmo Yang, and Eunbyung Park. Parameter-efficient instance-adaptive neural video compression. In *Proceedings of the Asian Conference on Computer Vision*, pages 250–267, 2024. 3

[40] Zirui Pan, Mengbai Xiao, Xu Han, Dongxiao Yu, Guanghui Zhang, and Yao Liu. patchdpcc: A patchwise deep compression framework for dynamic point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4406–4414, 2024. 2

[41] Jiahao Pang, Muhammad Asad Lodhi, and Dong Tian. Grasp-net: Geometric residual analysis and synthesis for point cloud compression. In *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis*, pages 11–19, 2022. 2

[42] Zizheng Que, Guo Lu, and Dong Xu. Voxelcontext-net: An octree based framework for point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6042–6051, 2021. 3

[43] Hongning Ruan, Yulin Shao, Qianqian Yang, Liang Zhao, and Dusit Niyato. Point cloud compression with implicit neural representations: A unified framework. In *2024 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 1709–1714. IEEE, 2024. 2, 3

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

[45] Rui Song, Chunyang Fu, Shan Liu, and Ge Li. Efficient hierarchical entropy model for learned point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14368–14377, 2023. 3

[46] Chang Sun, Hui Yuan, Shiqi Jiang, Da Ai, Wei Zhang, and Raouf Hamzaoui. Lpcm: Learning-based predictive coding for lidar point cloud compression. *arXiv preprint arXiv:2505.20059*, 2025. 3

[47] Haotian Tang, Shang Yang, Zhijian Liu, Ke Hong, Zhongming Yu, Xiuyu Li, Guohao Dai, Yu Wang, and Song Han. Torchsparse++: Efficient training and inference framework for sparse convolution on gpus. In *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2023. 6

[48] Koki Tsubota, Hiroaki Akutsu, and Kiyoharu Aizawa. Universal deep image compression via content-adaptive optimization with adapters. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2529–2538, 2023. 3

[49] Ties Van Rozendaal, Iris AM Huijben, and Taco S Cohen. Overfitting for fun and profit: Instance-adaptive data compression. *arXiv preprint arXiv:2101.08687*, 2021. 3

[50] Jianqiang Wang, Dandan Ding, Zhu Li, and Zhan Ma. Multiscale point cloud geometry compression. In *2021 Data Compression Conference (DCC)*, pages 73–82. IEEE, 2021. 2

[51] Jianqiang Wang, Dandan Ding, Zhu Li, Xiaoxing Feng, Chuntong Cao, and Zhan Ma. Sparse tensor-based multiscale representation for point cloud geometry compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 3, 6

[52] Jianqiang Wang, Ruixiang Xue, Jiaxin Li, Dandan Ding, Yi Lin, and Zhan Ma. A versatile point cloud compressor using universal multiscale conditional coding–part i: Geometry. *IEEE transactions on pattern analysis and machine intelligence*, 2024. 1, 2, 3, 6

[53] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 3

[54] Kangli Wang and Wei Gao. Unipcgc: Towards practical point cloud geometry compression via an efficient unified approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(12):12721–12729, 2025. 1, 2, 3

[55] Kangli Wang, Shihao Li, Qianxi Yi, and Wei Gao. A novel benchmark and dataset for efficient 3d gaussian splatting with gaussian point cloud compression. *arXiv preprint arXiv:2505.18197*, 2025. 3, 6

[56] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020. 3

[57] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 3

[58] Xinjie Wang, Yifan Zhang, Ting Liu, Xinpu Liu, Ke Xu, Jianwei Wan, Yulan Guo, and Hanyun Wang. Topnet: Transformer-efficient occupancy prediction network for octree-structured point cloud geometry compression. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 27305–27314, 2025. 3, 6

[59] Simon Wiedemann, Heiner Kirchhoffer, Stefan Matlage, Paul Haase, Arturo Marban, Talmaj Marinč, David Neumann, Tung Nguyen, Heiko Schwarz, Thomas Wiegand, et al. Deepcabac: A universal compression algorithm for deep neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):700–714, 2020. 6

[60] Louis Wiesmann, Andres Milioto, Xieyuanli Chen, Cyrill Stachniss, and Jens Behley. Deep compression for dense point cloud maps. *IEEE Robotics and Automation Letters*, 6(2):2060–2067, 2021. 3

[61] Xinju Wu, Pingping Zhang, Meng Wang, Peilin Chen, Shiqi Wang, and Sam Kwong. Geometric prior based deep human point cloud geometry compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(9):8794–8807, 2024. 2

[62] Shuting Xia, Tingyu Fan, Yiling Xu, Jenq-Neng Hwang, and Zhu Li. Learning dynamic point cloud compression via hierarchical inter-frame block matching. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7993–8003, 2023.

[63] Hao Xu, Xi Zhang, and Xiaolin Wu. Fast point cloud geometry compression with context-based residual coding and inr-based refinement. In *European Conference on Computer Vision*, pages 270–288. Springer, 2024. 2

[64] Ruixiang Xue, Jiaxin Li, Tong Chen, Dandan Ding, Xun Cao, and Zhan Ma. Neri: Implicit neural representation of lidar point cloud using range image sequence. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8020–8024. IEEE, 2024. 3

[65] Kang You, Kai Liu, Li Yu, Pan Gao, and Dandan Ding. Pointsoup: High-performance and extremely low-decoding-latency learned geometry codec for large-scale point cloud scenes. *arXiv preprint arXiv:2404.13550*, 2024. 2

[66] Kang You, Tong Chen, Dandan Ding, M Salman Asif, and Zhan Ma. Reno: Real-time neural compression for 3d lidar point clouds. *arXiv preprint arXiv:2503.12382*, 2025. 1, 2, 3, 6

[67] Pengpeng Yu, Haoran Li, Runqing Jiang, Jing Wang, Liang Lin, and Yulan Guo. Re-densification meets cross-scale propagation: Real-time neural compression of lidar point clouds. *arXiv preprint arXiv:2508.20466*, 2025. 3

[68] Pengpeng Yu, Ye Zhang, Fan Liang, Haoran Li, and Yulan Guo. Hierarchical distortion learning for fast lossy compression of point clouds. *IEEE Transactions on Multimedia*, 2025. 2

[69] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2021)*, 2021. 6

[70] Chenhao Zhang and Wei Gao. Adadpcc: Adaptive rate control and rate-distortion-complexity optimization for dynamic point cloud compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13188–13196, 2025. 2

[71] Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang, Hongwei Qin, Guo Lu, Jing Geng, and Jun Zhang. Gaussianimage: 1000 fps image representation and compression by 2d gaussian splatting. In *European Conference on Computer Vision*, pages 327–345. Springer, 2024. 3

[72] Yichi Zhang and Qianqian Yang. Efficient implicit neural compression of point clouds via learnable activation in latent space. *arXiv preprint arXiv:2504.14471*, 2025. 3

[73] Huiming Zheng, Wei Gao, Zhuozhen Yu, Tiesong Zhao, and Ge Li. Viewpcgc: view-guided learned point cloud geometry compression. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7152–7161, 2024. 2