Maniflat3D: Learning 3D Geometry Through Planar Representations from Multi-layer Unwrapping

Zijian Cao[1,2], Dayou Zhang[3*], Zeyuan Liu[2], Zhicheng Liang[1,2], Fangxin Wang[1,2,4*]

[1]School of Science and Engineering (SSE), The Chinese University of Hong Kong, Shenzhen
[2]Shenzhen Future Network of Intelligence Institute (FNii)
[3]College of Information Engineering, Capital Normal University
[4]The Guangdong Provincial Key Laboratory of Future Networks of Intelligence
{zijiancao1@link.,zhichengliang1@link.,wangfangxin@}cuhk.edu.cn, zhangdayou@cnu.edu.cn, lzy113367@gmail.com

Abstract

Point-based geometric representations such as point clouds and Gaussian Splatting are fundamental for 3D understanding. However, the inherent irregularity and high-dimensional nature of point structures present significant challenges for direct 3D learning approaches, which often struggle with scalability and achieve suboptimal performance due to sparse data distributions. In contrast, 2D learning paradigms benefit from well-established architectures with superior optimization stability and efficiency. To bridge this gap, we propose Maniflat3D, a unified framework that systematically transforms volumetric point-based geometries into structured 2D representations through a two-stage process: a multi-layer Ball-Pivoting reconstruction with adaptive density control, followed by Scalable Locally Injective Mapping (SLIM) to produce distortion-minimized, bijective UV parameterizations. Our approach explicitly encodes both geometric and attribute information into the flattened domain, enabling conventional 2D neural networks to effectively learn from complex 3D structures such as Gaussian Splatting. Experiments on the ShapeSplat dataset demonstrate that Maniflat3D achieves comparable performance while reducing parameter count by 90% compared to native 3D baselines, and simultaneously attains 21× compression ratio through neural encoding. These results establish a new paradigm for efficient geometric understanding, demonstrating successful transfer of planar learning advantages to challenging 3D manifold problems through dimensional reduction.

Introduction

Three-dimensional geometric understanding has become a cornerstone of modern computer vision and graphics, with applications in autonomous driving, robotics, and augmented reality. Point-based representations, particularly point clouds and 3D Gaussian Splatting (3DGS)(Kerbl et al. 2023), have gained significant traction due to their flexibility in capturing complex 3D structures and high-fidelity representation of geometric and appearance information.

However, when applied to learning-intensive tasks such as 3D scene classification and understanding, the irregular, unordered, and non-uniform nature of point-based representations poses significant challenges for neural networks designed for structured, grid-like data. These characteristics complicate feature extraction and optimization processes, necessitating specialized architectures such as PointNet (Qi et al. 2017), which frequently encounter limitations in computing efficiency, convergence stability and scalability.

In stark contrast, 2D neural networks have achieved remarkable success in learning-intensive tasks, benefiting from decades of architectural innovations and well-established training methodologies. The structured nature of 2D representations enables effective exploitation of spatial locality and facilitates cross-view feature learning. Rather than developing increasingly complex 3D-specific architectures, we propose to harness the proven effectiveness of 2D learning by systematically transforming point-based geometry into structured planar representations, thereby unlocking the potential of established 2D neural architectures for 3D understanding tasks.

The core of this transformation lies in *preserving strong correspondences between volumetric and planar structures while maintaining the fidelity of both geometric and semantic information*. Spatial occlusions, distance distortions, and the layering of interior and exterior surfaces pose significant challenges to the reasonable mapping of 3D representations onto structured grids, and existing works struggle to address these issues simultaneously. Multi-view projection methods (Feng et al. 2018; Yu et al. 2020; Hamdi, Giancola, and Ghanem 2021) suffer from information loss and occlusion artifacts, while surface parameterization techniques (Floater and Reimers 2001; Zhang et al. 2010) face difficulties with irregular non-manifold point clouds and Gaussian splatting.

To address these issues, we introduce **Maniflat3D**, a unified framework that systematically transforms volumetric point-based geometry into structured 2D representations through a carefully designed two-stage process. Our approach begins with an enhanced multi-layer Ball-Pivoting reconstruction augmented with adaptive density control, which robustly converts irregular point structures into manifold surfaces while preserving geometric details under varying point densities. Subsequently, we employ Scalable Locally Injective Mappings (SLIM) (Rabinovich et al. 2017) to generate UV coordinates with minimized distortion, main-

*Corresponding Authors.

taining geometric correspondence.

Maniflat3D injects geometric information and appearance attributes (color, covariance, opacity) from 3D objects into a flattened space, creating bijective sparse grids. This encoding strategy constructs structured representations compatible with 2D neural networks while preserving rich 3D semantic content, enabling efficient cross-domain transfer learning.

Our contributions can be summarized as follows:

- **A novel two-stage transformation paradigm Maniflat3D** that bridges the gap between irregular 3D point-based representation and planar grids, enabling the application of mature 2D architectures to complex 3D geometric understanding tasks.
- **Comprehensive application examples** that employ learning-based methods for object-level Gaussians processing. The novel neural encoding achieves a 21× compression ratio, and the sparse learning-based understanding framework enables effective 3D shape classification.
- **Extensive experimental validation** on the ShapeSplat dataset, demonstrating substantial improvements in computational efficiency and classification accuracy compared to native 3D baselines, thereby validating the effectiveness of Maniflat3D.

Related Work

3D Representations & Parameterization

The representation and parameterization of 3D geometry encompasses various approaches from classical projection methods to modern learning-based techniques. Traditional spherical projections (Cheng et al. 2021) provide mathematically well-defined mappings from 3D surfaces to 2D domains, though they often suffer from distortion artifacts at projection boundaries, while Bird's Eye View (BEV) (Lin et al. 2024) projections have gained popularity in autonomous driving and robotics for their intuitive top-down scene representations. Classical surface parameterization methods such as As-Rigid-As-Possible (Igarashi, Moscovich, and Hughes 2005) and Scalable Locally Injective Mappings (Rabinovich et al. 2017) focus on minimizing distortion while preserving geometric properties during 3D-to-2D mapping, albeit requiring explicit mesh representations. Recent learning-based approaches have explored 2D parameterizations of 3D signals without explicit triangle meshes, including UV mapping-based appearance models for humans and faces (Deng et al. 2018), and single-chart surface mappings for geometry processing tasks such as establishing surface correspondences (Srinivasan et al. 2024).

Point-based Deep Learning

Point-based Deep Learning encompasses two main paradigms. Direct methods operate on raw point cloud structures, evolving from foundational MLP-based architectures (Ma et al. 2022; Chen et al. 2023) like PointNet to more sophisticated approaches. Convolution-based methods (Wu, Qi, and Fuxin 2019; Thomas et al. 2019) address local correlation modeling through flexible operators, while transformer-based approaches (Wu et al. 2022, 2024) leverage self-attention mechanisms for capturing long-range dependencies. Recent developments focus on masked autoencoder methods (Zhang et al. 2022; Ren et al. 2024) and self-supervised pre-training strategies with improved robustness.

Projection-based methods transform 3D point clouds into 2D representations to leverage mature CNN architectures. Multi-view methods (Feng et al. 2018; Yu et al. 2020; Hamdi, Giancola, and Ghanem 2021; Yu and Song 2024) aggregate features from multiple rendered views, with subsequent works improving view relationships through advanced learning techniques. While these projection-based approaches effectively reduce 3D learning complexity, they face inherent limitations in maintaining complete spatial information during the projection process.

Method

As illustrated in Fig. 1, we propose a unified framework for compression and understanding of 3D point-based object geometry. Unlike convex hull surface point clouds that form smooth manifolds, complex irregular point-based geometries like Gaussian Splatting pose significant challenges for planar mapping due to their spatially non-uniform distributions and inability to be characterized as a single coherent manifold. To address the planar projection of such complex point-based geometries, we innovatively design a Multi-layer Ball-Pivoting Reconstruction and Distortion-Minimized UV Unwrapping approach. Furthermore, we employ a plug-and-play autoencoder to efficiently compress our planar grids, while mature 2D neural network frameworks effectively capture attribute information expressed in pixel form, thereby accomplishing effective 3D reasoning.

Preliminaries: 3DGS represents an object or a scene with a collection of Gaussians primitives (Liang et al. 2025) to model the geometry and view-dependent appearance (Zhang et al. 2025a). For a 3DGS set, $G = \{g_i\}_{i=1}^{N}$, representing an object with N individual Gaussians, the geometry of the i th Gaussian is explicitly parameterized via 3D covariance matrix Σ_i and its central coordinates $\sigma_i \in \mathbb{R}^3$ as:

$$g_i(x) = \exp\left(-\frac{1}{2}(x-\sigma_i)^T\Sigma_i^{-1}(x-\sigma)\right) \qquad (1)$$

where the covariance matrix $\Sigma_i = r_i s_i s_i^T r_i^T$ is factorized into a rotation matrix $r_i \in \mathbb{R}^4$ and a scale matrix $s_i \in \mathbb{R}^3$.

The visual appearance of the i^{th} Gaussian is parameterized by a set of Spherical Harmonics (SH) coefficients $f_i \in \mathbb{R}^{48}$ and an opacity value $o_i \in \mathbb{R}$. Thus, a single Gaussian can be represented by a set of five attributes as $g_i = \{\sigma_i, r_i, s_i, o_i, f_i\} \in \mathbb{R}^{14}$,and the entire 3DGS can be represented by a set of N such Gaussians as: $G = \{\{\sigma_i, r_i, s_i, o_i, f_i\}\}_{i=1}^{N}$.

Adaptive Multi-layer BPA-based Mapping

The discrete nature of Gaussian Splatting in space often fails to satisfy the continuity, smoothness, and local Euclidean properties required for manifold unfolding. However, dense Gaussians locally exhibit capturable manifold

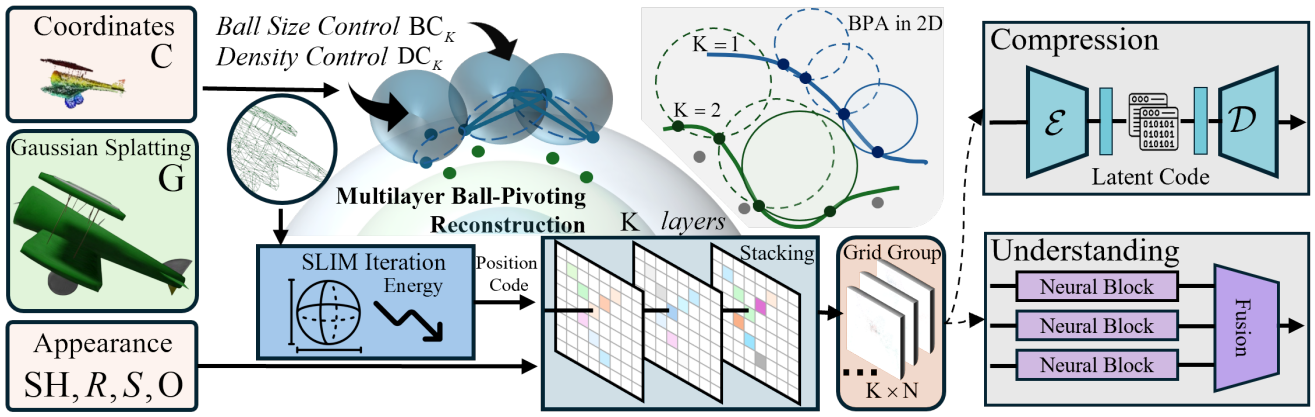

Figure 1: Our proposed Maniflat3D. For any object-level Gaussian Splatting, we perform multi-layer ball pivoting reconstruction on its coordinates to capture geometric information, and employ SLIM (Scalable Locally Injective Mapping) for minimal distortion unwrapping. The resulting unwrapped grid groups contain rich 3D geometric semantics, which can provide effective guidance for compression and understanding tasks.

features and can be regarded as compositions of multiple underlying manifolds. Assume the Gaussian point set $C = \{\sigma_i\}_{i=1}^{N}$ decomposes into K low-dimensional manifolds $M = \{m_j\}_{j=1}^{K}$, where each $\sigma_i \in m_j$. To automatically accomplish manifold separation, we propose an Adaptive Multi-layer Ball-Pivoting Algorithm-based mapping approach (see Alg. 1), inspired by surface reconstruction.

The Ball-Pivoting Algorithm (BPA) (Bernardini et al. 2002) is a geometric surface reconstruction technique that progressively constructs triangular meshes by simulating a virtual sphere rolling across point cloud surfaces. By systematically pivoting around edges and detecting valid triangulations through sphere-point intersections, BPA effectively transforms discrete point data into continuous surface representations. Building upon this foundation, our method utilizes multi-scale balls to automatically achieve layered decomposition through collision detection with Gaussians, followed by Scalable Locally Injective Mapping (SLIM) to minimize geometric distortion for each manifold layer, ultimately obtaining efficient planar mappings.

Adaptive Density Control. Executing BPA on irregular 3D Gaussians presents significant challenges, as severely non-uniform distributions and sparse representations make it difficult for the rolling ball trajectory to form smooth surfaces. To effectively distribute Gaussians across multiple low-dimensional manifolds, our approach aims to achieve uniform spatial distribution of Gaussians, followed by automatic extraction of smooth surfaces through ball pivoting.

We first employ a density controller (DC_K) to filter the original Gaussians and obtain uniform distribution. Initially, kernel density estimation is applied to all Gaussians, followed by adaptive downsampling of high-density regions according to a proportional factor β, thereby generating multi-layered relatively uniform manifolds. In cases where Gaussians are highly dispersed and cannot be adequately addressed by the density controller alone, we introduce compensatory connective elements to maintain the global 3D

Algorithm 1: MultiLayerBPA

Require: C (coordinates), DC_K (density control), BC_K (ball control), K (max layers)
Ensure: $M = \{m_j\}_{j=1}^{K}$

U $\leftarrow C, M \leftarrow \emptyset$
for $j = 1$ **to** K **do**
  $P \leftarrow DC$.filter(Unused, j)
  $r \leftarrow BC$.get_radius(j)
  if $len(P) <$ min_num **then**
    break
  end if
  $F \leftarrow \emptyset, m_j \leftarrow \emptyset$ {Initialize active front and mesh}
  while true **do**
    while $(e =$ get_active_edge$(F))$ **do**
      if $(\alpha =$ ball_pivot$(e, r))$ **and** (not_used$(\alpha) \vee$ on_front$(\alpha))$ **then**
        $m_j \leftarrow m_j \cup \{$triangle$(\alpha_i, \alpha, \alpha_j)\}$
        join(e, α, F)
        glue_if_exists(F)
      else
        mark_boundary(e)
      end if
    end while
    if triangle $=$ find_seed(P, r) **then**
      $m_j \leftarrow m_j \cup \{$triangle$\}$
      insert_edges(triangle, F)
    else
      break
    end if
  end while
  $M \leftarrow M \cup \{m_j\}$
  U $\leftarrow$ U $-$ extract_used(m_j)
end for
return M

structure, analogous to support structures in 3D printing.

Multilayer Ball-Pivoting Reconstruction. After obtaining a relatively uniform Gaussians distribution, we employ BPA to extract smooth manifold surfaces. As the virtual sphere collides with Gaussian points throughout the space, it continuously generates triangular mesh elements, which serve as the geometric foundation for our algorithm's surface capture. Note that BPA is an explicit surface reconstruction method that does not modify the original coordinates.

We introduce Ball Size Control (BC_K) to initialize the ball radius for each iteration round, thereby enhancing adaptability to varying density distributions across different layers. Previous studies have demonstrated that multi-scale ball-pivoting reconstruction can effectively mitigate reconstruction failures caused by non-uniform density distributions. Building upon this foundation, our BC_K mechanism additionally incorporates adaptive adjustments based on the density characteristics of each individual layer. During the initial iterations, where Gaussian point densities are relatively high, we employ smaller ball radius sets to ensure effective collision detection, resulting in the generation of numerous fine-grained and densely packed triangular facets. As the iterative process progresses to handle sparser layers, the ball radii are gradually increased to accommodate the reduced point density. It should be noted that the triangular patches generated in this process are not required to form a closed mesh topology. However, since excessive edge complexity can significantly complicate UV parameterization procedures, we incorporate hole-filling strategies to optimize the mesh structure and enhance its suitability for subsequent processing.

Geometric Global and Local Mapping. We design distortion-minimized unwrapping to allocate the Gaussians attached to each smooth manifold layer, where these points closely resemble vertices in a mesh structure. We employ Scalable Locally Injective Mapping (SLIM) for iterative optimization to establish a global-to-local, fine-grained projection mapping with minimized geometric distortion. Principal Component Analysis (PCA) (Maćkiewicz and Ratajczak 1993) is utilized as the initialization for UV unwrapping to accelerate the iterative process. The SLIM algorithm provides a robust framework for UV parameterization by minimizing distortion while ensuring local injectivity. The core objective function combines distortion energy and barrier terms in the following energy functional:

$$E(u) = \sum_{t\in T} A_t \left(\alpha \|J_t - R_t\|_F^2 + (1-\alpha)\frac{\|J_t\|_F^2}{\det(J_t)} \right) + \lambda \sum_{t\in T} B(\det(J_t)) \tag{2}$$

where u represents the UV coordinates, T is the set of triangles, A_t is the area of triangle t, J_t is the 2×2 Jacobian matrix of the parameterization, R_t is the closest rotation matrix to J_t, $\alpha \in [0, 1]$ balances conformal and authalic distortion, and $B(\cdot)$ is a barrier function that approaches infinity as $\det(J_t) \to 0^+$.

The symmetric Dirichlet energy term decomposes into two components measuring different distortion types. The first term $\|J_t - R_t\|_F^2$ penalizes conformal distortion by measuring deviation from the closest rotation, while the second term $\frac{\|J_t\|_F^2}{\det(J_t)}$ measures area distortion. To ensure local injectivity, the algorithm incorporates a logarithmic barrier function: $B(\det(J_t)) = -\log(\det(J_t))$. This barrier function prevents triangle flipping by approaching positive infinity as $\det(J_t)$ approaches zero.

The optimization employs an alternating minimization scheme that iterates between local and global phases. In the local phase, for each triangle t, the algorithm computes the optimal rotation matrix:

$$R_t^{(k)} = \arg\min_{R\in SO(2)} \|J_t^{(k)} - R\|_F^2 \tag{3}$$

which can be efficiently solved using SVD decomposition. In the global phase, the algorithm solves a linearized quadratic system:

$$\left(\sum_{t\in T} A_t W_t^{(k)} G_t^T G_t + \lambda H_{\text{barrier}}^{(k)} \right) u^{(k+1)} = \sum_{t\in T} A_t W_t^{(k)} G_t^T \text{vec}(R_t^{(k)}) \tag{4}$$

where G_t is the gradient operator matrix and the adaptive weights are defined as:

$$W_t^{(k)} = \alpha I + (1-\alpha)\frac{2I}{\det(J_t^{(k)})} \tag{5}$$

These weights ensure triangles with smaller determinants receive higher penalties.

Gaussian Attributes Stacking. To facilitate the fusion of Gaussian appearance attributes with geometry, we design a pixel-wise Gaussian attribute injection and stacking mechanism. Since the planar positions obtained through SLIM are in floating-point format, we quantize these positions to convert them into pixel coordinates, thereby transforming the attributes into colors within the planar image. We broadcast the pixel coordinates to all Gaussian attributes, with every three attributes forming the three channels of an image. All attribute maps with consistent pixel coordinates can be regarded as a series of planar grid groups.Since the number of Gaussians attached to different manifold layers varies, we implement a gradually decaying grid scale to maximize pixel space utilization.

Due to the minimal-distortion UV unwrapping iteration, neighboring pixels in the planar space typically correspond to adjacent regions in the actual 3D space. Consequently, Gaussian coordinates and DC spherical harmonics often exhibit smooth gradients.

Compression

Our compression framework leverages the inherent spatial coherence of planar representations through a three-stage pipeline: semantic-aware sorting, VQVAE-based quantization, and entropy coding. This approach systematically reduces information redundancy while preserving crucial geometric structure, as illustrated in Fig. 2.

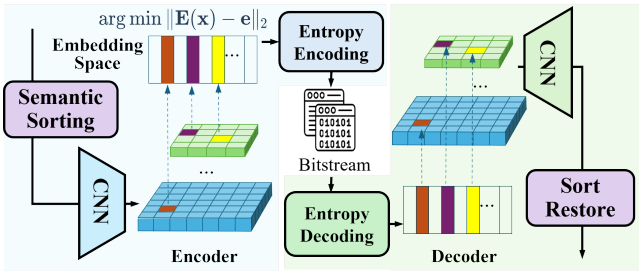

Figure 2: Overview of our three-stage compression pipeline: (a) Semantic-aware spatial rearrangement for coherent pixel layouts, (b) VQVAE encoding with multi-level codebooks, and (c) Brotli entropy coding.

Semantic-Aware Spatial Rearrangement. The flattened manifold representation preserves local geometry but lacks global coherence required for effective compression. We introduce a semantic-aware sorting algorithm that rearranges pixels based on perceptual similarity, creating a more compression-friendly representation.

Given a set of flattened manifold layers $\{L_j\}_{j=1}^{J}$ where each layer contains numerous colored pixels representing Gaussian attributes, we first extract all non-white pixels and compute a mapping function:

$$\Phi : (i, (y, x)) \mapsto (y', x') \tag{6}$$

where i is the layer index, (y, x) is the original pixel coordinate, and (y', x') is the new coordinate in the sorted representation. The mapping is constructed by sorting pixels based on HSV color space's hue value:

$$H(p_1) < H(p_2) \Rightarrow \Phi(p_1) < \Phi(p_2) \tag{7}$$

This operation creates a visualization where perceptually similar attributes are placed in proximity, enhancing spatial coherence. The source-to-target pixel mapping is stored in a compact grayscale image where intensity values encode layer indices, enabling efficient reconstruction.

Vector Quantized Variational Autoencoder. To further compress the sorted representation, we employ a Vector Quantized Variational Autoencoder (VQVAE) (Van Den Oord, Vinyals et al. 2017) that effectively captures recurring patterns in our flattened manifold space. The VQVAE consists of an encoder E, a decoder G, and a codebook $\mathcal{Z} = \{z_k\}_{k=1}^{K}$ with K discrete codes.

The encoder maps input images x to latent representations $E(x)$, which are then quantized by finding the nearest codebook vector:

$$q(z|x) = \begin{cases} 1, & \text{if } z = \arg\min_{z_k \in \mathcal{Z}} \|E(x) - z_k\|_2 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

The decoder then reconstructs the input from these quantized representations. Training minimizes:

$$\mathcal{L} = \|x - G(z_q)\|_2^2 + \beta\|sg[E(x)] - z_q\|_2^2 + \gamma\|sg[z_q] - E(x)\|_2^2 \tag{9}$$

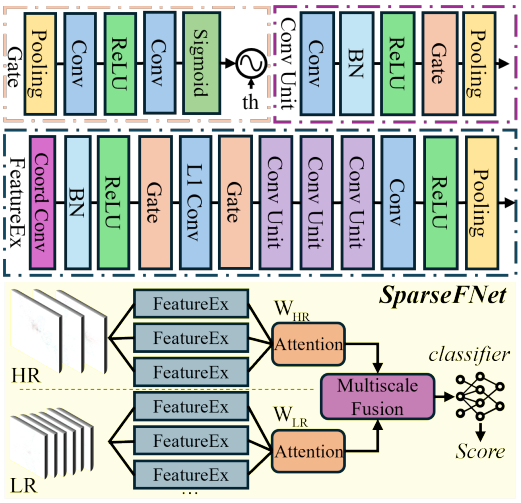

Figure 3: Our proposed SparseFNet. This network is used to validate that our proposed novel planar representation can provide effective 3D geometric semantics for spatial understanding.

where z_q is the quantized code, $sg[\cdot]$ is the stop-gradient operator, and β, γ are weighting coefficients.

Our implementation features a multi-scale structure that captures both fine details and global structure. The encoder produces multi-level indices $\{ind_i\}_{i=1}^{L}$ that reference entries in corresponding codebooks, enabling high-fidelity reconstruction while requiring significantly less storage.

Entropy Coding with Brotli. The VQVAE indices exhibit strong statistical patterns that can be further exploited through entropy coding. We apply Brotli (Alakuijala et al. 2018) compression, which combines LZ77 (Kreft and Navarro 2010) dictionary-based compression with a context-aware prefix coding similar to Huffman encoding.

Brotli performs particularly well on our quantized indices due to the semantic sorting step creating long runs of similar values, the VQVAE codebook structure introducing predictable patterns, and the multi-level index representation containing hierarchical redundancy. This synergy between our preprocessing pipeline and Brotli's compression mechanisms yields optimal results for our specific distribution.

Understanding

Our proposed representation is rich in geometric semantic features, providing knowledge for neural networks to achieve both global understanding and local analysis of 3D data. The extensive body of work emerging in 2D deep learning can be leveraged to develop our planar representation, including CNNs (LeCun et al. 1989), ViT (Dosovitskiy et al. 2020), and diffusion models (Ho, Jain, and Abbeel 2020). Since our framework generates sparse colored pixels, Sparse Learning becomes crucial for its development and utilization. Sparse Learning (Wen et al. 2016) focuses on handling data with numerous zero elements, employing components such as sparse convolution (Liu et al. 2015)

and sparse regularization (Selesnick 2017) techniques. Empowered by these components, our framework theoretically supports object classification and recognition, semantic segmentation, object synthesis and completion, demonstrating significant application potential and scalability.

By invoking and combining Sparse Learning components, we have validated that planar grids can adequately correspond to the original geometric semantics. This correspondence not only preserves the structural information of the original 3D data but also significantly reduces computational complexity through sparse representation, making 3D scene processing on IoT devices feasible. For simplicity, we employ a Gaussian object recognition task here, which effectively validates the effectiveness of our framework.

We developed a Sparse Multilayer Fusion Network (SparseFNet) to leverage our planar representation, with the network architecture shown in Fig. 3. Following the layered approach described above, Gaussian objects are processed to obtain planar grids containing a small portion of high-resolution (HR) images and a majority of low-resolution (LR) images. HR images better represent the distribution of Gaussians attached to the outer manifold, while LR images typically provide high-curvature details and internal filling. The images are fed into feature extraction layers, followed by attention modules that capture feature importance across different layers, formulated as:

$$F_{out} = \sum_{i=1}^{N} W_i \times F_i \tag{10}$$

Subsequently, the high-dimensional semantic vectors from HR and LR undergo multi-scale fusion. Since this differs from the different resolution fusion in low-level vision, we choose to use linear layers to approximate this connection. A classifier composed of fully connected layers produces the final results from these high-dimensional semantic vectors.

The FeatureEx architecture used for feature extraction consists of sparse learning and classical 2D deep learning components. First, CoordConv (Liu et al. 2018) is employed to integrate planar positional coordinate information into the post-convolution tensors, ensuring that original sparse pixels can be perceived by the neural network. The introduction of L1-regularized convolutional neural networks serves the same purpose. In the subsequent architecture, we introduce Gate modules that set channel components below a threshold to zero after a series of processing steps, reducing unnecessary semantic redundancy. Gate modules can also be integrated within Conv Units. All module designs are guided by sparse learning theory, not only effectively extracting rich geometric semantic knowledge from planar grid groups but also significantly reducing computational complexity.

Experiment

Experimental Details

Mapping Details. We have carefully designed the parameters for our 3D point-based representation transformation. The initial ball size is set to 0.005, with incremental growth of at least 0.002 per iteration or adaptive adjustment based on inter-point distances. For the adaptive density control mechanism, we employ a density estimation-guided threshold design, applying uniform downsampling with a factor of β (typically 0.3) to regions exceeding the threshold. The initial ball sizes below 0.001 cause mesh failures, but within 0.003-0.006, performance differences are minimal - Chamfer Distance fluctuations remain within 1%.

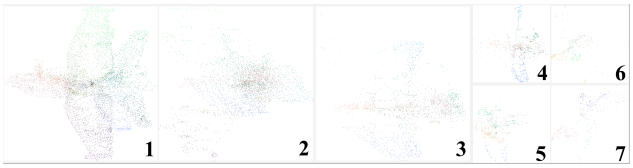

Figure 4: Adaptive multi-layer BPA-based mapping details (first 7 layers): A multi-resolution encoding scheme is adopted, where each pixel's RGB values correspond to three attribute parameters of a Gaussian Splatting point.

The generated planar representation layers exhibit varying resolutions: the first three layers are configured as HR (512×512), while the remaining layers use LR(256×256). These settings are optimized to accommodate Gaussian Splatting scenarios ranging from ten thousand to 100K Gaussians. We set the maximum expansion layer count $K = 10$ to ensure coverage of 99% of the Gaussians.

Prior to injecting Gaussian attributes into the planar representation, normalization and quantization preprocessing is required. We conducted a comprehensive analysis examining how the compression of different Gaussian attributes affects visual quality, incorporating findings from relevant literature (Zhang et al. 2025b; Morgenstern et al. 2024). Based on these findings, we established our quantization scheme: 16-bit quantization for coordinate information and 8-bit quantization for other attributes. The quantized attributes can be encoded into image color channels, with multi-layer visualizations shown in Fig. 4.

Dataset. We employ the ShapeSplat dataset for 3D Gaussian Splatting, which comprises 65K objects spanning 87 distinct categories. Based on this data, we construct a comprehensive surface capture dataset containing over 8K diverse Gaussian objects from 8 categories to rigorously evaluate the effectiveness and robustness of our 3D framework.

Compression Experimental Setup. The compression ratio testing experiments were conducted following the unwrapping process. During the testing phase, we utilized code from the original 3DGS implementation (Kerbl et al. 2023) and portions from the 3DGS_PoseRender framework (Liu and Mass 2023). We fine-tuned our VQVAE model (Beyer* et al. 2024) specifically for our dataset and task requirements. All the training and testing were conducted on a Linux server equipped with an Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz and an NVIDIA A100-PCIE-40GB GPU.

Understanding Experimental Setup. To demonstrate the efficacy of our approach, we focus on the fundamental task of shape semantic classification as our primary validation benchmark. Our training configuration employs a learning rate of 0.0004, cross-entropy loss as the objective function,

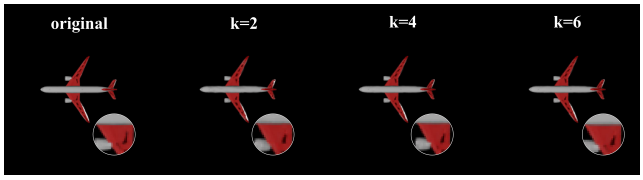

Figure 5: Hierarchical visual effects are demonstrated through samples with $k = 2, 4, 6$. For Gaussian Splatting at the scale of 30K points, $k = 2$ already captures the core semantic information, despite exhibiting color discontinuities and geometric distortions; when $k > 5$, the reconstruction achieves near-complete coverage of the primary Gaussians, with only negligible differences.

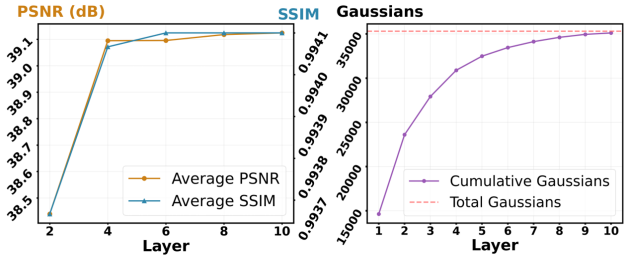

Figure 6: Quantitative analysis of multiple layers and Gaussian accumulation curves.

and processes data with a batch size of 8.

Experimental Results

Hierarchical Mapping Analysis. For our proposed novel mapping scheme, we conduct a detailed analysis of the hierarchical relationships. To facilitate demonstration and understanding, we select Gaussian objects with 20K-60K Gaussians for evaluation. The visual contribution of each layer is visualized as shown in Fig. 5. Large-scale layer Gaussians capture primary visual semantics, while small-scale layers refine details. Using only the first two layers causes color discontinuities and pattern distortions, but retains sufficient semantic content for downstream tasks. We also report corresponding quantitative analysis, demonstrating the relationship between visual quality (PSNR and SSIM (Wang et al. 2004)) and cumulative layer count, as illustrated in Fig. 6. The corresponding conclusions and visualization results mutually validate each other.

Compression. Our novel representation enables effective compression of Gaussians. Unlike other Gaussian compression tasks, we adopt the rendering quality of the original Gaussians as our evaluation benchmark. According to our experiments, by retaining only the top 5 layers with higher visual contribution, our method achieves a 21× compression ratio with acceptable visual quality degradation (PSNR loss of approximately 2.8 dB).

Understanding. For Gaussian shape classification, direct learning baselines on Gaussians are extremely rare; therefore, we predominantly adopt point cloud learning paradigms for comparison. However, we have adapted the original network architectures to accommodate Gaussians, ensuring no inherent bias or inequality in the evaluation. Our classification task thoroughly validates that the proposed novel representation provides sufficient geometric and semantic information, and that the designed sparse representation network can effectively capture the corresponding features. The model parameters and the mean accuracy are reported in the Table 1.

Method	Data Type	Parameters	mAcc
PointNet	Gaussians*	1.65M	0.8534
PointConv	Gaussians*	19.56M	0.8219
PTv2	Gaussians	11.3M	0.8821
PointMLP	Gaussians*	12.6M	0.9138
DeLA	Gaussians	5.33M	0.9014
SparseFNet	Maniflat3D	**630K**	0.8858

Table 1: Comparison of Gaussian semantic classification performance. Our method achieves comparable performance to numerous baselines while significantly reducing model parameters. * indicates that downsampling was applied.

Our planar pixel grid representation provides effective sparse knowledge for neural networks. When processing complete Gaussian Splatting data, we achieve comparable performance using an architecture with only 5% of the parameters required by 3D deep learning networks. The mature sparse learning paradigm endows our planar representation with tremendous potential for further development. Most methods that directly learn 3D Gaussian representations face significantly higher computational overhead, which poses a non-negligible burden for the democratization of 3D object understanding applications.

Conclusion

We present Maniflat3D, a novel framework that bridges 3D point-based representations and structured 2D learning paradigms. Our two-stage pipeline, combining enhanced multi-layer Ball-Pivoting reconstruction with SLIM parameterization, effectively converts complex 3D structures into bijective UV mappings with minimal distortion. Experimental validation on the ShapeSplat dataset demonstrates the framework's effectiveness, achieving comparable performance with significantly reduced model parameters compared to native 3D baselines, while accomplishing a 21× compression ratio of Gaussians through neural coding.

Maniflat3D demonstrates the potential for applying established 2D learning architectures to advance 3D geometric understanding. By addressing the computational limitations of specialized 3D architectures, our framework enables leveraging mature 2D methodologies for complex 3D tasks. In future work, we intend to investigate diffusion models to expand applications encompassing Gaussian object inpainting and generation. This work highlights the promise of flattened 3D representations and may inspire further exploration in this direction, offering valuable contributions to autonomous driving and virtual reality technologies.

Acknowledgements

The work was supported in part by the Guangdong S&T Programme with Grant No. 2024B0101030002, the Basic Research Project No. HZQB-KCZYZ-2021067 of Hetao Shenzhen-HK S&T Cooperation Zone, the NSFC with Grant No. 62293482 and Grant No. 62471423, the Shenzhen Science and Technology Program with Grant No. JCYJ20241202124021028 and Grant No. JCYJ20230807114204010, the Guangdong Special Support Program (Grant No. 2024TQ08X346), the Shenzhen Outstanding Talents Training Fund 202002, the Young Elite Scientists Sponsorship Program of CAST (Grant No. 2022QNRC001), the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001) and the Shenzhen Key Laboratory of Big Data and Artificial Intelligence (Grant No. SYSPG20241211173853027). The work of Dayou Zhang was supported in part by "Ecological Immersive Learning Project" of Tianjin Normal University (Grant No. 2025JGDS016Y) and the Scientific and Technological Innovation Service Capacity Building Project (Grant No. 25534510007).

References

Alakuijala, J.; Farruggia, A.; Ferragina, P.; Kliuchnikov, E.; Obryk, R.; Szabadka, Z.; and Vandevenne, L. 2018. Brotli: A general-purpose data compressor. *ACM Transactions on Information Systems (TOIS)*, 37(1): 1–30.

Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; and Taubin, G. 2002. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4): 349–359.

Beyer*, L.; Steiner*, A.; Pinto*, A. S.; Kolesnikov*, A.; Wang*, X.; Salz, D.; Neumann, M.; Alabdulmohsin, I.; Tschannen, M.; Bugliarello, E.; Unterthiner, T.; Keysers, D.; Koppula, S.; Liu, F.; Grycner, A.; Gritsenko, A.; Houlsby, N.; Kumar, M.; Rong, K.; Eisenschlos, J.; Kabra, R.; Bauer, M.; Bošnjak, M.; Chen, X.; Minderer, M.; Voigtlaender, P.; Bica, I.; Balazevic, I.; Puigcerver, J.; Papalampidi, P.; Henaff, O.; Xiong, X.; Soricut, R.; Harmsen, J.; and Zhai*, X. 2024. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*.

Chen, B.; Xia, Y.; Zang, Y.; Wang, C.; and Li, J. 2023. Decoupled local aggregation for point cloud learning. *arXiv preprint arXiv:2308.16532*.

Cheng, A.-C.; Li, X.; Sun, M.; Yang, M.-H.; and Liu, S. 2021. Learning 3d dense correspondence via canonical point autoencoder. *Advances in Neural Information Processing Systems*, 34: 6608–6620.

Deng, J.; Cheng, S.; Xue, N.; Zhou, Y.; and Zafeiriou, S. 2018. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7093–7102.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Feng, Y.; Zhang, Z.; Zhao, X.; Ji, R.; and Gao, Y. 2018. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 264–272.

Floater, M. S.; and Reimers, M. 2001. Meshless parameterization and surface reconstruction. *Computer Aided Geometric Design*, 18(2): 77–92.

Hamdi, A.; Giancola, S.; and Ghanem, B. 2021. Mvtn: Multi-view transformation network for 3d shape recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1–11.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Igarashi, T.; Moscovich, T.; and Hughes, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)*, 24(3): 1134–1141.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4): 139–1.

Kreft, S.; and Navarro, G. 2010. LZ77-like compression with fast random access. In *Data Compression Conference*, 239–248. IEEE.

LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; and Jackel, L. D. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4): 541–551.

Liang, Z.; Zhang, D.; Shen, L.; Zhang, M.; Zhang, J.; Ju, B.; Dasari, M.; Wang, F.; and Liu, J. 2025. 4DGStream: Variable Bitrate Dynamic Gaussian Splatting Streaming. *lEEE Transactions on Multimedia*.

Lin, Z.; Wang, Y.; Qi, S.; Dong, N.; and Yang, M.-H. 2024. Bev-mae: Bird's eye view masked autoencoders for point cloud pre-training in autonomous driving scenarios. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 3531–3539.

Liu, B.; Wang, M.; Foroosh, H.; Tappen, M.; and Pensky, M. 2015. Sparse convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 806–814.

Liu, G.; and Mass, G. 2023. 3DGS_PoseRender: A Framework for 3D Gaussian Splatting Pose Rendering. GitHub repository.

Liu, R.; Lehman, J.; Molino, P.; Petroski Such, F.; Frank, E.; Sergeev, A.; and Yosinski, J. 2018. An intriguing failing of convolutional neural networks and the coordconv solution. *Advances in neural information processing systems*, 31.

Ma, X.; Qin, C.; You, H.; Ran, H.; and Fu, Y. 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *arXiv preprint arXiv:2202.07123*.

Maćkiewicz, A.; and Ratajczak, W. 1993. Principal components analysis (PCA). *Computers & Geosciences*, 19(3): 303–342.

Morgenstern, W.; Barthel, F.; Hilsmann, A.; and Eisert, P. 2024. Compact 3d scene representation via self-organizing gaussian grids. In *European Conference on Computer Vision*, 18–34. Springer.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Rabinovich, M.; Poranne, R.; Panozzo, D.; and Sorkine-Hornung, O. 2017. Scalable locally injective mappings. *ACM Transactions on Graphics (TOG)*, 36(4): 1.

Ren, B.; Mei, G.; Paudel, D. P.; Wang, W.; Li, Y.; Liu, M.; Cucchiara, R.; Van Gool, L.; and Sebe, N. 2024. Bringing Masked Autoencoders Explicit Contrastive Properties for Point Cloud Self-Supervised Learning. In *Proceedings of the Asian Conference on Computer Vision*, 2034–2052.

Selesnick, I. 2017. Sparse regularization via convex analysis. *IEEE Transactions on Signal Processing*, 65(17): 4481–4494.

Srinivasan, P. P.; Garbin, S. J.; Verbin, D.; Barron, J. T.; and Mildenhall, B. 2024. Nuvo: Neural uv mapping for unruly 3d representations. In *European Conference on Computer Vision*, 18–34. Springer.

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Wen, W.; Wu, C.; Wang, Y.; Chen, Y.; and Li, H. 2016. Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 29.

Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 9621–9630.

Wu, X.; Jiang, L.; Wang, P.-S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; and Zhao, H. 2024. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4840–4851.

Wu, X.; Lao, Y.; Jiang, L.; Liu, X.; and Zhao, H. 2022. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35: 33330–33342.

Yu, H.-T.; and Song, M. 2024. Mm-point: Multi-view information-enhanced multi-modal self-supervised 3d point cloud understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 6773–6781.

Yu, Q.; Yang, C.; Fan, H.; and Wei, H. 2020. Latent-MVCNN: 3D shape recognition using multiple views from pre-defined or random viewpoints. *Neural Processing Letters*, 52(1): 581–602.

Zhang, D.; Liang, Z.; Cao, Z.; Wang, D.; and Wang, F. 2025a. SRBF-Gaussian: Streaming-Optimized 3D Gaussian Splatting. In *Proceedings of the IEEE Conference Virtual Reality and 3D User Interfaces*, 461–471.

Zhang, D.; Liang, Z.; Cao, Z.; Wei, L.; Wang, D.; and Wang, F. 2025b. 3dgstreaming: Spatial heterogeneity aware 3D gaussian splatting compression and streaming. *IEEE Internet of Things Journal*, 12(20): 41625 – 41636.

Zhang, L.; Liu, L.; Gotsman, C.; and Huang, H. 2010. Mesh reconstruction by meshless denoising and parameterization. *Computers & Graphics*, 34(3): 198–208.

Zhang, R.; Guo, Z.; Gao, P.; Fang, R.; Zhao, B.; Wang, D.; Qiao, Y.; and Li, H. 2022. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems*, 35: 27061–27074.