Geometric Deep Learning for 3D Shape Representation: A Comparative Study

Report

Akansh Patel
S2 Labs

December 25, 2025

Executive Summary

Three-dimensional data representations such as point clouds, voxel grids, meshes, and implicit surfaces are widely used in scientific and engineering applications. However, conventional deep learning models, designed for Euclidean grid-structured data, are often ill-suited for processing such irregular geometric domains. Geometric Deep Learning (GDL) addresses this limitation by incorporating geometric priors and enabling learning directly on non-Euclidean structures.

This report presents a comparative study of common 3D shape representations and their associated neural architectures for latent shape learning. Both theoretical properties and practical autoencoder-based experiments are analyzed to evaluate reconstruction quality, computational efficiency, and modeling flexibility.

The study demonstrates that the choice of representation is inherently task-dependent, involving trade-offs among accuracy, efficiency, and scalability, and provides practical guidance for selecting 3D representations suited to specific tasks and computational constraints.

Contents

1 Introduction

Many real-world machine learning problems involve data that reside in geometric domains such as graphs, manifolds, and groups, rather than unstructured vector spaces. In 3D shape learning, such data are commonly represented as point clouds, voxel grids, surface meshes, or continuous implicit functions. Using the geometric structure of these representations effectively is crucial for robust generalization.

Although classical deep learning models perform well on Euclidean data, they are not well suited for general geometric domains. Naively vectorizing structured objects compromises topological integrity, limiting sample efficiency. Moreover, standard architectures often lack inductive biases, such as symmetry, locality, and multiscale structure, which are essential for capturing complex shape interactions [1].

Geometric Deep Learning (GDL) addresses these challenges by providing a principled framework for learning on non-Euclidean domains. By modeling data as signals on structured spaces and explicitly incorporating geometric priors such as equivariance and stability, GDL enables efficient learning on complex geometric data.

The choice of 3D representation point clouds, voxel grids, meshes, or implicit surfaces is a fundamental design decision, as each representation captures different geometric aspects and involves distinct trade-offs in terms of computational cost, expressiveness, and learning behavior. In this report, we implement autoencoder models for point clouds, voxels, and implicit surfaces, and evaluate their reconstruction performance. Mesh-based methods are discussed conceptually due to implementation constraints. These experiments provide practical insights into how representation choice affects efficiency, reconstruction quality, and overall usability within the Geometric Deep Learning framework.

1.1 Scope

This report focuses on the experimental evaluation of different 3D representations for shape learning using autoencoder models. Point clouds, voxel grids, and implicit surfaces are implemented and tested, while mesh-based methods are discussed conceptually. This study emphasizes reconstruction quality, computational efficiency, and trade-offs in selecting appropriate geometric representations.

2 Objectives

The objectives of this work are the following.

- To study the principles of Geometric Deep Learning for 3D data.
- To understand multiple 3D representations.
- To implement autoencoder models for different 3D formats.
- To compare reconstruction quality and model behavior.

3 Experiments and Results

This section describes the datasets, 3D representations, and autoencoder architectures used in the experimental evaluation.

3.1 Dataset Description and Preprocessing

The experiments in this study use the **MaizeField3D** dataset [2], which consists of 1,045 high-quality 3D point clouds of field-grown maize plants acquired using terrestrial laser scanning (TLS). Each shape is represented by a point cloud of 10,000 points, obtained through uniform subsampling to balance geometric fidelity and computational efficiency. All point clouds are normalized and aligned to a common coordinate frame prior to training. The dataset exhibits complex plant geometry with fine structural details, making it well suited for evaluating autoencoder architectures and assessing reconstruction performance on realistic 3D point cloud data.

3.2 3D Geometric Representations and Models

This section presents the 3D geometric representations considered in this study and the corresponding neural architectures used to learn latent shape representations.

3.2.1 Point Cloud

Point clouds represent 3D shapes as unordered sets of points. Memory usage scales linearly with the number of points; higher resolutions provide finer detail but increase computational cost. Point clouds do not explicitly capture surface topology. Consequently, processing is typically facilitated by permutation-invariant operators, such as those utilized in the PointNet [3] and PointNet++ [4] architectures.

Preprocessing: Each point cloud is downsampled to 2,048 points using Farthest Point Sampling and normalized to a common coordinate frame. The processed data are stored for efficient training.

PointNet++ Implementation: The architecture employs a PointNet++ encoder to map input point clouds into a 128-dimensional latent representation, followed by a fully connected MLP decoder for shape reconstruction. As illustrated in Figure 1a, the encoder performs hierarchical sampling, neighborhood grouping, and local feature extraction to capture both local and global geometric features. The model is optimized using the bidirectional (symmetric) Chamfer Distance loss to enforce spatial correspondence. Training is conducted with a batch size of 8 using the Adam optimizer at a learning rate of 1×10^{-3}.

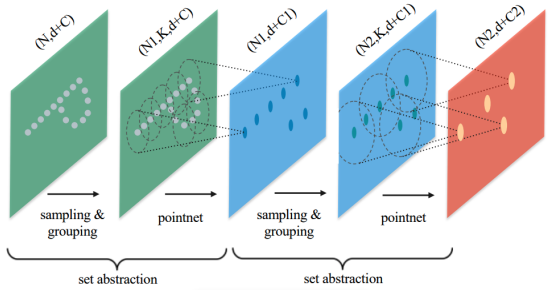
Results: The PointNet++ autoencoder achieved a final bidirectional Chamfer Distance loss of 0.032 after 3,500 training epochs, with a total training time exceeding 30 hours. Qualitative reconstruction results are shown in Figure 1b, where the reconstructed point clouds capture the overall global structure of the maize plants with moderate accuracy, while only partially recovering fine-grained leaf geometry due to the inherent complexity of thin, highly articulated plant shapes, which are challenging to reconstruct reliably.

3.2.2 Voxel

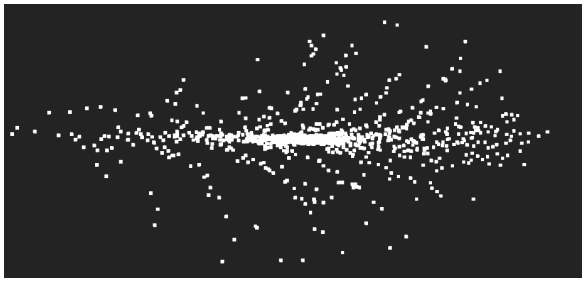
Voxels extend the concept of pixels to 3D by discretizing volumes into regular cubic grids. This regular structure encodes spatial adjacency, enabling efficient 3D convolutional operations; however, memory complexity scales cubically with resolution ($O(N^3)$), limiting high-resolution processing. Despite this, voxels are effective for volumetric segmentation and dense data learning.

Preprocessing: Point clouds are normalized and discretized into a fixed-resolution $128 \times 128 \times 128$ voxel occupancy grid for input to a 3D CNN–based autoencoder.

3D CNN Implementation: The encoder employs 3D convolutions and the decoder uses transposed convolutions, with a 128-dimensional latent space. The model is trained with MSE loss, batch size 16, Adam optimizer, and learning rate 1×10^{-3} for 50 epochs. The architecture is illustrated in Figure 2a.

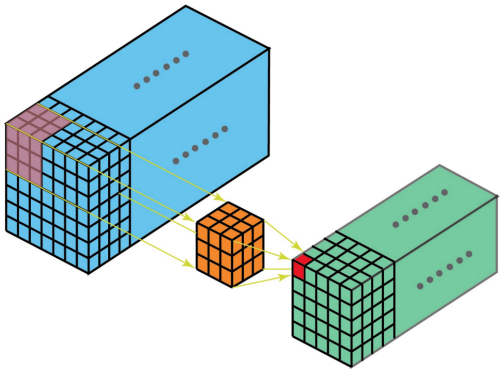


(a) Hierarchical sampling, grouping, and feature extraction in PointNet++ [4].

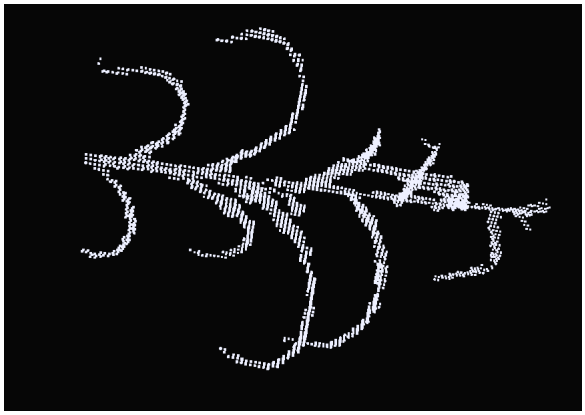


(b) Reconstructed maize plant point cloud using PointNet++ autoencoder. The model captures global shape accurately but shows moderate loss of thin leaves.

Figure 1: Illustration of the PointNet++ autoencoder for point cloud data. (a) shows the hierarchical feature extraction process, and (b) displays a reconstructed point cloud from the MaizeField3D dataset.



(a) 3D convolution operation in the autoencoder (adapted from [5]).



(b) Voxel-based reconstruction at 128^3 resolution. Main canopy volume is preserved, but thin leaves are lost and blocky artifacts appear due to limited resolution.

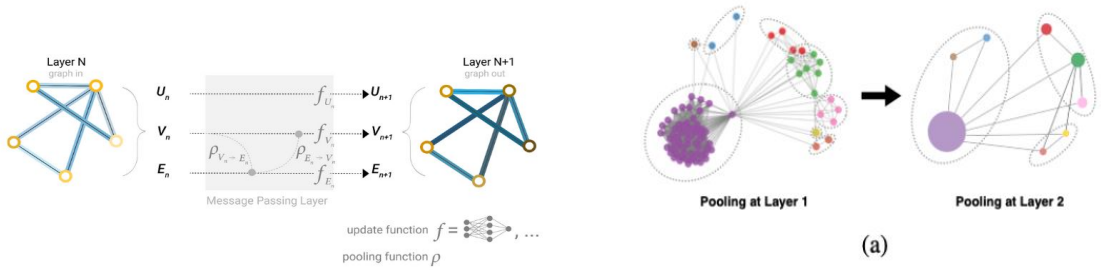
Figure 2: Illustration of the 3D CNN autoencoder for voxel data. (a) shows the 3D convolution operation in the encoder, and (b) displays a reconstructed voxel from the MaizeField3D dataset.

Results: The 3D CNN autoencoder converged rapidly, achieving a Mean Squared Error (MSE) loss of 0.0012 within 30 minutes of training. Qualitative evaluation, as shown in Figure 2b, indicates that the model captures the global volumetric structure of the maize plants with moderate fidelity, while exhibiting smoothing and blocky artifacts that lead to the loss of thin leaves and fine-scale details, and reflect resolution limitations as well as sensitivity to input orientation and scale.

3.2.3 Mesh

Meshes compactly represent 3D surfaces using vertices, edges, and faces, enabling adaptive refinement and graph or topology-aware operations. Triangular meshes are common and widely used in computer graphics, CAD, and simulation tasks for representing detailed surfaces. However, generating high-quality meshes with arbitrary topology is challenging, and computational complexity is higher than for volumetric representations.

Theoretical Analysis: Mesh-based representations can be effectively processed using graph neural networks (GNNs) with edge pooling to preserve geometric details and enable



(a) Triangular mesh processing using a Graph Neural Network (GNN) (adapted from [6]).

(b) Edge pooling operation for hierarchical graph reduction.

Figure 3: Illustration of mesh processing with GNNs and edge pooling. (a) shows how a triangular mesh is converted into a graph and processed using a GNN, while (b) demonstrates edge pooling to reduce graph size while preserving geometric features.

shape reconstruction. Triangular meshes are first converted into graph structures, where vertices serve as nodes, triangle edges define graph connections, and node features may include coordinates, normals, or curvature. During GNN processing, nodes aggregate information from their neighbors through message passing, capturing both local and global mesh structure. Edge pooling (Figure 3b) hierarchically merges connected nodes, reducing graph size while retaining essential geometric information in the latent representation. While such approaches are theoretically promising, they remain computationally expensive. The architecture of the GNN with edge pooling is illustrated in Figure 3a.

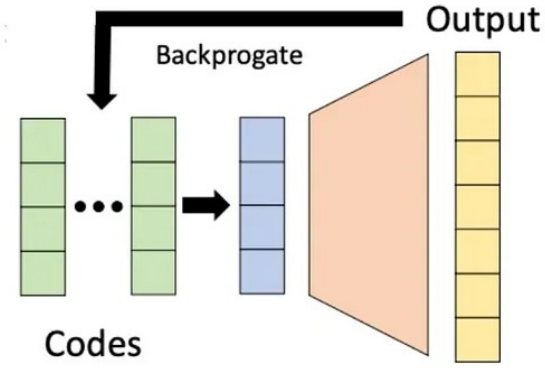
3.2.4 Signed Distance Field (SDF)

Signed Distance Fields (SDFs) provide a continuous, resolution-independent representation of 3D shapes by encoding the distance of any point in space to the closest surface, with the surface defined implicitly as the zero level set. SDFs are parameterized by a function, often learned by a neural network, enabling implicit surface modeling without discretization and with moderate computational cost for function evaluation and sampling. They are memory-efficient, facilitate the generation of shapes with arbitrary topology, and can be easily converted to other 3D representations such as meshes or voxel grids, making them a versatile choice for geometry learning.

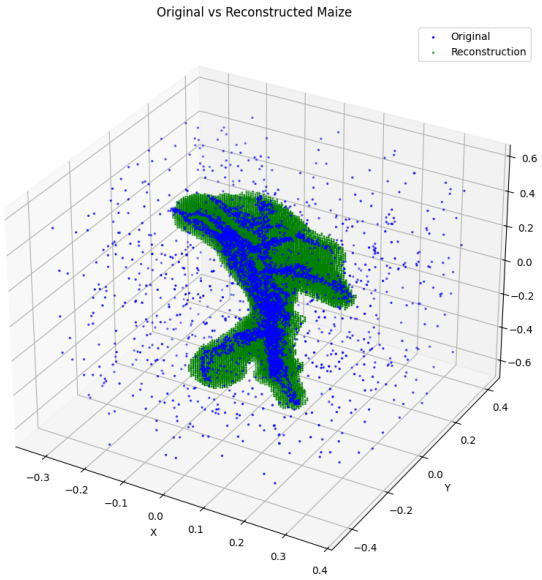
Preprocessing: Meshes are normalized and converted into point–SDF pairs by sampling 5,000 spatial points per shape (80% near surface, 20% uniform) and computing signed distances via nearest-surface and inside–outside tests.

DeepSDF Implementation: DeepSDF [7] follows an auto-decoder architecture in which each shape is represented by a learnable 128-dimensional latent code. A shared 8-layer MLP decoder with a hidden dimension of 256 and skip connections predicts signed distance values from the concatenation of latent codes and 3D query points, as illustrated in Figure 4a. Training is performed using an ℓ_1 SDF loss with latent code regularization to prevent overfitting. The model is optimized using the Adam optimizer with a learning rate of 5×10^{-4} and a batch size of 32, with SDF values clamped to ± 0.1 to improve numerical stability.

Results: The DeepSDF model was trained for 5,000 epochs using 128-dimensional latent codes, reaching a final SDF loss of 0.0015 after approximately 2.5 hours of training. As shown in Figure 4b, the reconstruction captures the global canopy geometry and major branches of the maize plant, demonstrating effective implicit representation learning with good global fidelity. Fine-scale leaf tips and thin structures are moderately oversmoothed. Skip connections improve training stability, and extended training or denser near-surface sampling could further enhance



(a) DeepSDF Auto decoder, at time of inference [7]



(b) DeepSDF reconstruction of maize plant. Green indicates the reconstructed surface, blue shows the original point cloud. Global structure is well preserved, though fine leaves are moderately smoothed.

Figure 4: Qualitative results of DeepSDF on shape reconstruction. (a) shows the auto-decoder during inference, and (b) compares the reconstructed shape with the original.

fidelity.

3.3 Summary of Reconstruction Performance

Reconstruction varies across representations: PointNet++ captures global shape but misses fine details, 3D CNNs are fast but blocky, and DeepSDF preserves overall structure with slight smoothing (Table 1).

Architecture	Resolution / Samples	Loss & Metric	Qualitative Reconstruction Summary
PointNet++	2,048 Points	Sym.Chamfer ≈ 0.032	Captures overall plant shape with moderate accuracy; thin leaves and fine details are only partially preserved.
3D CNN	128^3 grid	MSE ≈ 0.0012	Recovers main canopy volume, but shows blocky artifacts and loses slender leaf structures.
DeepSDF	5,000 samples/shape	ℓ_1 SDF ≈ 0.0015	Preserves global canopy and major branches well, while moderately oversmoothing thin leaves and tips.

Table 1: Comparison of reconstruction performance for PointNet++, 3D CNN, and DeepSDF autoencoders on MaizeField3D using a 128-dimensional latent space.

4 Key Takeaways

Based on experiments with the MaizeField3D dataset, the following observations can be made:

- **Representation Trade-offs:** Voxel grids are memory-intensive but straightforward to process, whereas point clouds and meshes offer more compact and flexible representations for capturing maize plant geometry.
- **Task-Oriented Representation Selection:** Representation choice should be guided by maize phenotyping objectives, desired structural detail, and computational efficiency requirements.
- **Mesh versus Point Cloud Learning:** Meshes preserve surface connectivity and topology for graph-based processing, while PointNet++ effectively extracts hierarchical geometric features from point clouds.
- **Implicit Representation Benefits:** DeepSDF provides continuous, resolution-independent representations, accurately modeling global canopy structure with moderate smoothing of fine leaf tips.
- **Reconstruction Constraints and Enhancements:** Preprocessing, downsampling, and limited sampling density can reduce fidelity for fine maize leaf structures, highlighting the potential of hybrid representations and data augmentation to improve generalization.

5 Conclusion

This report investigated the effectiveness of point cloud, voxel, and implicit Signed Distance Field representations for 3D shape learning within the Geometric Deep Learning framework, with mesh-based approaches analyzed conceptually due to implementation constraints. On the MaizeField3D dataset, the PointNet++ point cloud autoencoder provided a favorable balance between reconstruction quality and implementation complexity, capturing the overall plant shape with moderate global accuracy while only partially preserving thin leaves and fine-scale structure. The voxel-based 3D CNN autoencoder converged rapidly and recovered the main canopy volume, but exhibited blocky artifacts and loss of slender leaf structures owing to resolution limitations and sensitivity to input orientation and scale. The DeepSDF model achieved good alignment of the global canopy and major branches and demonstrated effective continuous shape modeling, while moderately oversmoothing thin components at the chosen sampling density and training budget. Overall, the results confirm that no single 3D representation is universally optimal; the choice should be driven by task requirements, tolerance for errors on thin structures, and computational constraints, motivating representation-aware model design and future exploration of hybrid and higher-capacity methods for high-fidelity reconstruction of complex plant geometries.

5.1 Limitations

The study has several limitations related to data, architecture, and computational constraints:

- **Sensitivity to Noise:** Point clouds, voxels, and SDFs are affected by input noise, which can distort features and reduce reconstruction fidelity.
- **Partial Transformation Invariance:** Models like PointNet++, 3D CNNs, and DeepSDF are not fully invariant to rotations, translations, or scaling, limiting generalization to unaligned inputs.
- **Preprocessing Artifacts:** Steps such as downsampling, normalization, voxelization, SDF sampling, and Marching Cubes may cause loss of fine details or smoothing of complex structures.

- **Data Resolution Limitations:** Reduced point cloud sizes or coarse SDF sampling improve efficiency but can miss small-scale structures and intricate geometry.
- **Limited Hybrid Exploration:** Only individual representations were studied; hybrid approaches (e.g., voxel + SDF) could improve fidelity and robustness.
- **Computational Cost and Training Time:** High-fidelity models for processing point cloud and mesh-based require significant resources and long training times.

6 Future Work

- Employ methods like **AtlasNet** [8] to directly generate mesh representations, improving surface fidelity and enabling end-to-end mesh learning.
- Use transformer-based models such as **Point-M2AE** [9] to enhance feature learning robustness for point clouds.
- Integrate **T-Net** [3] or similar alignment modules as preprocessing to make representations transformation-invariant before conversion to specific formats.
- Utilize high-resolution **VoxelNet** [10] or multi-resolution approaches to capture fine geometric details efficiently.
- Apply augmentation strategies such as random rotations, scaling, jittering, and noise injection to improve model robustness and generalization to unaligned or noisy inputs.
- Explore hybrid and implicit representations (e.g., voxel + SDF, occupancy networks) to leverage complementary strengths and enable high-fidelity reconstruction.

References

[1] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.

[2] Elvis Kimara, Mozhgan Hadadi, Jackson Godbersen, Aditya Balu, Talukder Jubery, Yawei Li, Adarsh Krishnamurthy, Patrick S Schnable, and Baskar Ganapathysubramanian. Agrifield3d: A curated 3d point cloud and procedural model dataset of field-grown maize from a diversity panel. *arXiv preprint arXiv:2503.07813*, 2025.

[3] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017.

[4] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[5] Shobhit Verma. When should i use 3d convolutions?, July 2019.

[6] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alex Wiltschko. A gentle introduction to graph neural networks. *Distill*, 6(8), August 2021.

[7] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[8] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation, 2018.

[9] Renrui Zhang, Ziyu Guo, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, Hongsheng Li, and Peng Gao. Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training, 2022.

[10] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection, 2017.