

PLATYPUS: Progressive Local Surface Estimator for Arbitrary-Scale Point Cloud Upsampling

Donghyun Kim¹, Hyeonkyeong Kwon², Yumin Kim¹, Seong Jae Hwang^{1*}

¹Yonsei University, Seoul, South Korea

²Korea University, Seoul, South Korea

{danny0103, yumin, seongjae}@yonsei.ac.kr, rhaxod9513@korea.ac.kr

Abstract

3D point clouds are increasingly vital for applications like autonomous driving and robotics, yet the raw data captured by sensors often suffer from noise and sparsity, creating challenges for downstream tasks. Consequently, point cloud upsampling becomes essential for improving density and uniformity, with recent approaches showing promise by projecting randomly generated query points onto the underlying surface of sparse point clouds. However, these methods often result in outliers, non-uniformity, and difficulties in handling regions with high curvature and intricate structures. In this work, we address these challenges by introducing the Progressive Local Surface Estimator (PLSE), which more effectively captures local features in complex regions through a curvature-based sampling technique that selectively targets high-curvature areas. Additionally, we incorporate a curriculum learning strategy that leverages the curvature distribution within the point cloud to naturally assess the sample difficulty, enabling curriculum learning on point cloud data for the first time. The experimental results demonstrate that our approach significantly outperforms existing methods, achieving high-quality, dense point clouds with superior accuracy and detail.

Introduction

Recently, autonomous driving, robotics, and other technologies that utilize 3D data have attracted significant interest, leading to the growing popularity of 3D point clouds as a representation of 3D data. However, the raw point clouds captured by sensors such as LiDAR, depth cameras often contain significant noise, and the distribution becomes particularly sparse for points that are farther from the sensor. Therefore, *point cloud upsampling*, the task of increasing the density points of a sparse 3D point cloud to be of dense (e.g., Fig. 1 sparse Input to dense GT) is vital for effectively using raw data in tasks like classification and segmentation, leading to various methods to tackle this challenge.

Starting with early works (Alexa et al. 2003; Lipman et al. 2007; Huang et al. 2009) that employed optimization-based methods, the rise of deep learning has led to the proposal of various learning-based methods (Yu et al. 2018; Li et al. 2019; Qian et al. 2021a), for training point cloud upsampling networks. Existing learning-based methods have

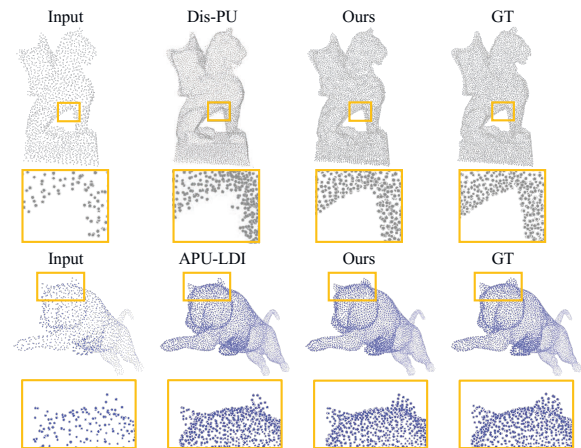


Figure 1: Comparative visualization of $4\times$ point cloud upsampling results on PU1K. Point cloud upsampling is the task of generating a denser point cloud (i.e., rightmost column) that accurately reflects the underlying geometry of a sparse point cloud (i.e., leftmost column). Our method successfully upsamples intricate areas where existing methods struggle to perform well.

demonstrated excellent performance but they have several drawbacks. These methods typically split the sparse input point cloud into multiple patches, upsample each patch, and then recombine them. This approach (i.e., *split-and-combine* process), which fails to consider the relationships between patches during upsampling, often results in issues such as holes, outliers, and non-uniformity, especially at the boundaries where the patches are combined.

To address the issues arising from the *split-and-combine* process, recent studies (He et al. 2023; Li et al. 2024) have proposed a pipeline that moves randomly generated points (i.e., query points) onto the surface that the point cloud inherently represents. In the upsampling process of this pipeline, initial query points are generated around the sparse point cloud (i.e., input point cloud). Next, these query points are projected onto the underlying surface of the sparse point cloud. To accurately determine the underlying surface, it is necessary to know the unsigned distance field of the ground-truth point cloud, but during upsampling, the ground-truth

*Corresponding author.

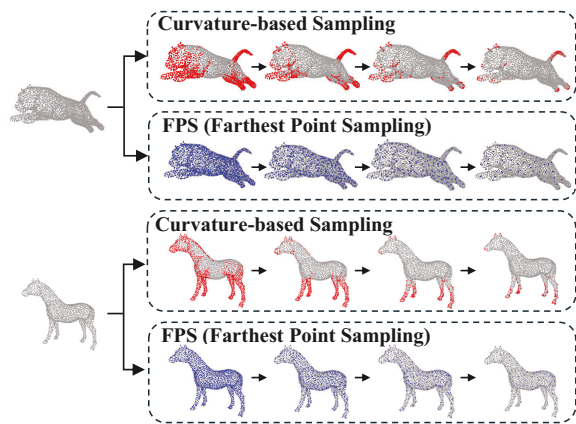


Figure 2: An analysis of the differences between FPS (Farthest Point Sampling) and our newly proposed curvature-based sampling. The comparison shows which points remain as the point cloud is progressively sampled down to fewer points using each sampling method. FPS uniformly samples points across the entire point cloud, whereas curvature-based sampling selectively samples points from regions with intricate structures and high curvature values.

point cloud is not available. Therefore, in the training phase, the network is trained to predict the unsigned distance from randomly generated query points around the sparse point cloud to its underlying surface. This approach enables the network to infer the underlying surface of the dense point cloud using only the sparse point cloud.

While this pipeline has addressed many of the issues in previous methods, there are still some significant problems that remain unresolved. As seen in Fig. 1, areas within the point cloud that require high curvature and locally complex structures (e.g., animal paws and ears, object edges) are often poorly upsampled. To better capture local features in complex regions that existing methods struggle to upsample, our work employs a novel approach using the concept of *curvature value* within the distance-estimating network of this pipeline. Curvature value represents the degree of curvature in a specific area, calculated by considering the geometric relationships with surrounding points. This metric allows us to quantitatively identify regions where existing methods fail to upsample effectively. We then utilize our newly proposed curvature-based sampling technique within the network to explicitly sample these regions with high curvature values (Fig. 2). We refer to this network as *Progressive Local Surface Estimator* (PLSE), which progressively retains regions with high curvature and intricate structures, enabling the network to focus on extracting features from these critical areas.

Using the curvature value, which effectively highlights regions in the point cloud where upsampling is challenging, we further enhance the learning process of PLSE by implementing a new curriculum learning strategy that calculates the difficulty of a point cloud based on the distribution of curvature values. As shown in Fig. 1, the network’s feature extractor generally struggles more with capturing features in local regions with complex structures and high curvature

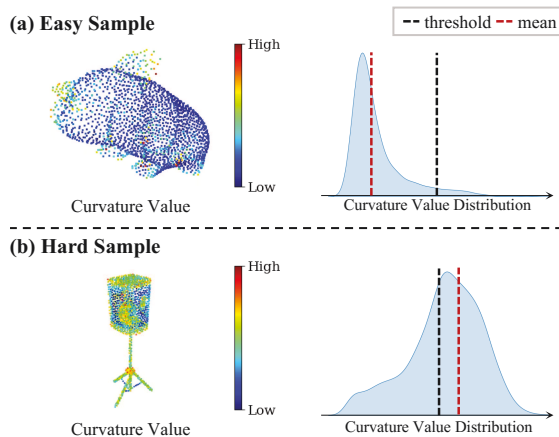


Figure 3: An analysis of the easy samples and hard samples used in our curriculum learning strategy. (a) Point clouds with a higher proportion of points with low curvature values, resulting in a distribution with high skewness, were classified as easy samples. (b) Conversely, point clouds with a higher proportion of points with high curvature values were classified as hard samples.

compared to simpler, flatter areas. Based on this observation, if a point cloud has fewer complex structures—indicated by a curvature value distribution skewed toward lower values and a lower mean—we classify it as an easy sample from a learning perspective (Fig. 3). Conversely, if the point cloud’s curvature value distribution is skewed toward higher values, indicating more complex structures, it is classified as a hard sample. Following the curriculum learning strategy, easy samples were used during the early epochs, while hard samples were introduced in the later epochs, helping the network to more effectively learn the unsigned distance field.

Contributions. In this work, we introduce *PLATYPUS* (**P**rogressive **L**ocal **S**urface **E**stimator for **A**rbitrary-**S**cale **P**oint **C**loud **U**p**S**ampling) for point cloud upsampling. Specifically, we make the following contributions:

- We propose a novel network, Progressive Local Surface Estimator (PLSE), to learn the unsigned distance field from sparse point clouds. PLSE employs a curvature-based sampling method, which allows our network to explicitly focus on extracting features from critical areas.
- To improve the learning of the unsigned distance field, we implement a curriculum learning strategy, which classifies training samples into easy and hard based on the skewness of curvature value distribution.
- The results from diverse experiments demonstrate that our approach achieves state-of-the-art performance.

Related Works

Optimization-based Point Cloud Upsampling. As a pioneering approach to point cloud upsampling, Alexa et al. 2003 proposed interpolating points at vertices of the Voronoi diagram in the local tangent space. Later, Lipman

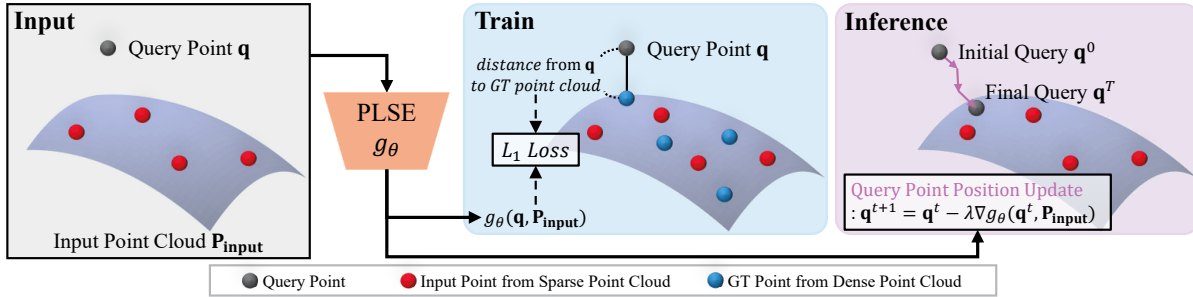


Figure 4: Overall Pipeline of PLATYPUS. *Input*: The input consists of a sparse point cloud $\mathbf{P}_{\text{input}}$ and nearby generated query points \mathbf{q} . *Train*: During the training process, our Progressive Local Surface Estimator (PLSE) g_θ is trained to predict the distance from the query point to the underlying surface of the sparse point cloud. The loss compares the distance from the query point \mathbf{q} to the surface of the sparse point cloud (assumed to be identical to the ground truth point cloud). *Inference*: the initial query point \mathbf{q}^0 is progressively updates using the PLSE gradient ∇g_θ to become the final query \mathbf{q}^T projected onto the surface of the input point cloud.

et al. 2007 introduced the locally optimal projection (LOP) operator for point resampling and surface reconstruction, which was further improved by Huang et al. 2009 with weighted LOP for iterative normal estimation. However, these LOP-based methods assume points are sampled from smooth surfaces, reducing upsampling quality near sharp edges and corners. To address this, an edge-aware resampling method was introduced, though it relies heavily on normal information and parameter tuning. Subsequently, Wu et al. 2015 proposed a method using Meso-skeleton guidance for consolidation and completion. Overall, optimization-based methods may fail when prior assumptions are unmet.

Learning-based Point Cloud Upsampling. With the rise of deep learning in point cloud analysis, learning-based methods have made significant breakthroughs in upsampling tasks. PU-Net (Yu et al. 2018) was the first learning-based method, introducing multi-scale feature learning per point and expanding point sets via multi-branch MLPs. However, it overlooked geometric structures, which led to the development of improved methods like the multi-step patch-based progressive network by MPU (Yifan et al. 2019) and PU-GAN (Li et al. 2019), which used generative adversarial networks to handle sparse, non-uniform inputs. PUGeoNet (Qian et al. 2020) introduced a geometry-centric upsampling network, while Dis-PU (Li et al. 2021) separated the upsampling task into a dense generator and spatial refiner. Building on these advancements, recent methods like NePs (Feng et al. 2022) utilized neural fields for high-resolution surface representation, and Grad-PU (He et al. 2023) decomposed upsampling into midpoint interpolation and location refinement. APU-LDI (Li et al. 2024) introduced an unsigned distance field guided by a local distance indicator for arbitrary-scale upsampling, while Rep-KPU (Rong et al. 2024) used kernel point deformation and cross-attention mechanisms. However, these methods still struggle with capturing local geometry exquisitely, and no methods yet have explored ways to explicitly locally measure the degree of geometric complexity.

Implicit Neural Representation. Implicit neural representation (INR) achieved great performance in 3D shape representation. Conventionally, implicit neural representation works with neural networks by approximating shape functions, such as signed distance functions (SDF) (Park et al. 2019) or unsigned distance functions (UDF) (Chibane, Pons-Moll et al. 2020). Recently, some research in point cloud upsampling has shown advantages by leveraging implicit neural representation for surface representation (Feng et al. 2022; Zhao et al. 2022, 2023; He et al. 2023; Li et al. 2024). Yet, these INR-based methods innately require per-sample training of the INR networks which quickly becomes impractical when the task demands a fast inference on unseen point clouds.

Methods

In this section, we provide a detailed description of our work, PLATYPUS. First, we explain the progression of our upsampling pipeline. Following that, we introduce our novel network, Progressive Local Surface Estimator (PLSE), and our curriculum learning strategy that leverages the distribution of curvature values.

Overall Pipeline: Projection-based Upsampling

The pipeline used by existing methods often generate outliers because the patches are upsampled independently without considering each other. To address this issue, we follow a projection-based pipeline (He et al. 2023; Li et al. 2024) that predicts the underlying surface of the point cloud and projects points onto the predicted surface (Fig. 4).

Upsampling Process. The upsampling process is as follows: query points are randomly generated around the sparse point cloud that needs upsampling. These generated query points are then projected onto the underlying surface of the sparse point cloud using a distance minimization process, similar to the approach used in Grad-PU (He et al. 2023). The distance minimization process utilizes the unsigned distance field of the ground-truth point cloud to move a point to its ground-truth position. For distance minimization, we

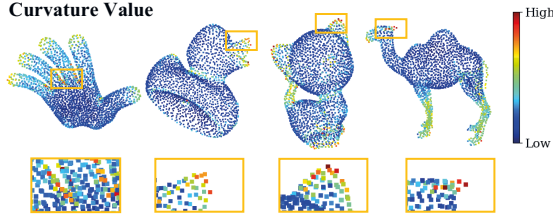


Figure 5: Visualization of each point’s curvature value. Points with high curvature values are shown in red, while points with low curvature values are shown in blue.

need a distance function $f(\cdot)$ that outputs the shortest distance from a point \mathbf{q} to the ground-truth point cloud. The distance minimization process can be described by the following equation:

$$\mathbf{q}^{t+1} = \mathbf{q}^t - \lambda \nabla f(\mathbf{q}^t, \mathbf{P}_{\text{gt}}), \quad t = 0, \dots, T - 1. \quad (1)$$

Here, $\mathbf{q}^t \in \mathbb{R}^3$ represents the position of the query point at iteration t , $\mathbf{P}_{\text{gt}} \in \mathbb{R}^{N \times 3}$ is the ground-truth point cloud of N points, λ is the step size, and $\nabla f(\mathbf{q}^t, \mathbf{P}_{\text{gt}})$ is the gradient of the distance function at \mathbf{q}^t . Following this approach, the initially generated query point \mathbf{q}^0 undergoes several iterations. As a result, the final projected point \mathbf{q}^T will be positioned on the underlying surface of the ground-truth point cloud.

Distance Function Training. However, during inference, the ground-truth point cloud \mathbf{P}_{gt} is unavailable, which restricts the use of the distance function $f(\cdot)$ for the distance minimization process. To address this issue, we train a network $g_\theta(\cdot)$ to predict the shortest distance from a point \mathbf{q} to the underlying surface of the ground-truth point cloud using only the point \mathbf{q} and the input point cloud $\mathbf{P}_{\text{input}}$ (i.e., sparse point cloud). Therefore, we train the network $g_\theta(\mathbf{q}, \mathbf{P}_{\text{input}})$ which essentially functions as the ideal distance function $f(\cdot)$ such that $f(\mathbf{q}, \mathbf{P}_{\text{gt}}) \approx g_\theta(\mathbf{q}, \mathbf{P}_{\text{input}})$. This, similar to Eq. (1), allows the gradient to be approximated using only $\mathbf{P}_{\text{input}}$ and iteratively projects the input point as follows:

$$\mathbf{q}^{t+1} = \mathbf{q}^t - \lambda \nabla g_\theta(\mathbf{q}^t, \mathbf{P}_{\text{input}}), \quad t = 0, \dots, T - 1. \quad (2)$$

Hence, in this projection-based upsampling framework, the quality of the network $g_\theta(\cdot)$ directly dictates the point cloud upsampling quality. In light of this, our work aims to develop a new network for $g_\theta(\cdot)$ which addresses the aforementioned challenges as we describe next.

Progressive Local Surface Estimator

We implement a novel network called Progressive Local Surface Estimator (PLSE), which serves the role of $g_\theta(\cdot)$. This network employs a method that effectively captures local features in regions with high curvature and intricate structures, which existing methods struggle to upsample accurately.

Curvature Value. We analyze that effectively upsampling challenging areas with intricate structures, such as the edges of objects and the ears of animals, requires focused learning of their local features. To achieve this, it is necessary

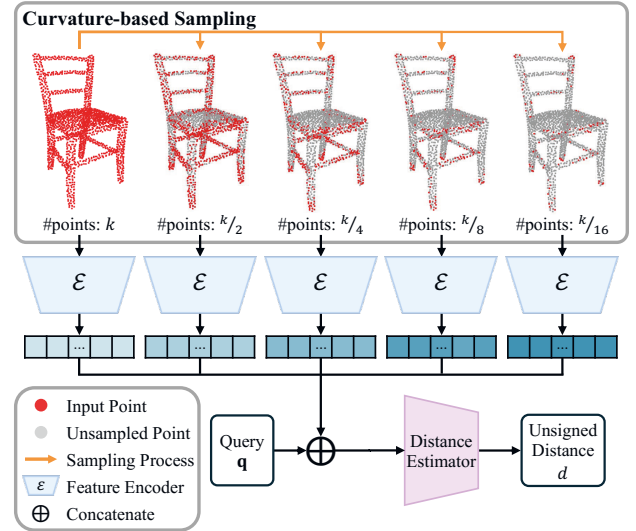


Figure 6: An illustration showing the structure of Progressive Local Surface Estimator (PLSE). The input point cloud with k points is sampled through curvature-based sampling into point clouds with $k/2$, $k/4$, $k/8$, and $k/16$ points. Each of these point clouds passes through a feature encoder to generate features. These features are then concatenated and passed along with the query through the distance estimator, which outputs the unsigned distance d .

to establish metrics and criteria that quantitatively define these regions. We find that these structurally complex regions have relatively *high curvature* compared to other areas within the point cloud. By applying the unsigned scalar of umbrella curvature (Foorginejad and Khalili 2014), we calculate the curvature value $c_{\mathbf{p}}$ for each point \mathbf{p} using the following equation:

$$c_{\mathbf{p}} = \frac{1}{K} \sum_{i=1}^K \left| \frac{\mathbf{x}_i}{|\mathbf{x}_i|} \cdot \hat{\mathbf{n}} \right|. \quad (3)$$

Here, \mathbf{x}_i is the vector from \mathbf{p} to its neighboring point \mathbf{p}_i (i.e., $\mathbf{x}_i = \mathbf{p}_i - \mathbf{p}$) for each of K neighboring points. $\hat{\mathbf{n}}$ is the surface normal vector at \mathbf{p} which is estimated using Open3D (Zhou, Park, and Koltun 2018) due to the absence of the ground-truth surface at \mathbf{p} . Intuitively, based on Eq. (3), points in regions with complex structures or sharp curvatures will have high curvature values, while points in less complex or flatter areas will have lower curvature values. As shown in Fig. 5, the calculated curvature values effectively represent the degree of structural complexity, and intricacy of each point in the point cloud.

Curvature-based Sampling. We first precompute the curvature value of each point as the criterion for assessing the complexity and detail of the structure. Then, we employ a novel technique called *curvature-based sampling* to ensure the network effectively captures local features in regions with complex structures. As shown in Fig. 6, curvature-based sampling explicitly selects points with relatively high curvature values from the sparse point cloud, *progressively*

sampling point clouds of various sizes over multiple iterations. Given a sparse point cloud with k points, we sample the top $k/2$ points with the highest curvature values in the first sampling step. In the subsequent steps, we sample the top $k/4$ points, followed by the top $k/8$ points, progressively focusing on regions with more complex structures.

Pipeline of PLSE. In our network, called Progressive Local Surface Estimator (PLSE), we extract features \mathbf{f}_0 through \mathbf{f}_4 from each of sampled point clouds, where each feature $\mathbf{f}_i \in \mathbb{R}^d$. The extracted features are then concatenated with the coordinates of the query point $\mathbf{q} \in \mathbb{R}^3$, resulting in a final feature vector $\mathbf{f}_{\text{final}} \in \mathbb{R}^{3+d \times 5}$. This final feature vector $\mathbf{f}_{\text{final}}$ is then passed through the distance estimator module, which outputs the predicted shortest distance from the query points to the underlying surface of the point cloud.

Loss Formulation. To ensure that the output of PLSE, the unsigned distance $g_\theta(\mathbf{q}, \mathbf{P}_{\text{input}})$, represents the shortest distance from the query point to the underlying surface of the ground-truth point cloud, PLSE optimizes the L1 loss between its output $g_\theta(\mathbf{q}, \mathbf{P}_{\text{input}})$ and $f(\mathbf{q}, \mathbf{P}_{\text{gt}})$ which is distance from the query point \mathbf{q} to the nearest point in the ground-truth point cloud \mathbf{P}_{gt} . This optimization ensures that PLSE accurately predicts the shortest distance from the query point to the underlying surface.

Curriculum Learning with Global Curvature Value

Thus far, we have been using the curvature value to locally characterize each point. However, we realize that the notion of curvature value may easily extend to characterize the global structure complexity of the entire point cloud as a new summary measure, namely, *global curvature value*. This point cloud-level notion of complexity fundamentally allows us to identify *easy samples* (i.e., simple point clouds of points with small c_p) and *hard samples* (i.e., complex point clouds of points with high c_p). Interestingly, this criterion naturally enables *curriculum learning* (Bengio et al. 2009; Duan et al. 2020) which is a strategy where the network learns samples in order of increasing difficulty, starting with *easy samples* and advancing to *hard samples*. Although curriculum learning is a widely used versatile scheme, we note that applying it to point cloud has not been straightforward due to the absence of sample difficulty measures on point clouds. In our work, our insights on curvature allows us to employ a curriculum learning strategy to aid in the training of our network PLSE.

Global Curvature Value. To obtain the global curvature value, which is a measure that effectively reflects global structure complexity, we calculate the curvature value for each point in the training sample (i.e., input point cloud) and analyze the distribution of these curvature values. Given a training sample $\mathbf{P}_{\text{input}}$ with N points, where each point \mathbf{p}_i has a curvature value $c_{\mathbf{p}_i}$, the global curvature value of $\mathbf{P}_{\text{input}}$ is calculated as follows:

$$\text{global curvature value} = \frac{1}{N} \sum_{i=1}^N c_{\mathbf{p}_i}. \quad (4)$$

We classify the sample’s difficulty based on whether the global curvature value is above or below the threshold we

Method	PU-GAN (4×)			PU-GAN (16×)		
	CD↓	HD↓	P2F↓	CD↓	HD↓	P2F↓
PU-Net	0.401	4.927	4.231	0.323	5.978	5.445
MPU	0.327	4.859	3.070	0.194	6.104	3.375
PU-GAN	0.281	4.603	3.176	0.172	5.237	3.217
Dis-PU	0.265	3.125	2.369	0.150	3.956	2.512
PU-GCN	0.268	3.201	2.489	0.161	4.283	2.632
NePs	0.385	5.615	1.642	0.147	8.851	1.925
Grad-PU	0.245	2.369	1.893	0.108	2.352	2.217
APU-LDI	0.232	1.675	1.338	0.092	1.504	1.544
PU-VoxelNet	0.233	1.751	2.137	0.091	1.726	2.301
RepKPU	0.248	2.880	1.906	0.107	3.345	2.068
PLATYPUS	0.229	1.426	1.908	0.088	1.429	2.142

Table 1: Quantitative comparisons against other methods on the PU-GAN dataset.

set. If a training sample has many points with high curvature values and few with low values, it will have a difficulty score above the threshold, classifying it as a hard sample. Conversely, if there are few points with high curvature values and many with low values, the sample will have a difficulty score below the threshold, classifying it as an easy sample. According to Eq. (3), each point can have a curvature value between 0 and 1, and we set the threshold at 0.5.

We use these classified easy samples during the early epochs of the training phase and hard samples during the late epochs to train PLSE. This approach achieves significant performance improvements for PLSE.

Experiments

Experimental Setup

Datasets. We conducted experiments using two synthetic datasets, PU-GAN (Li et al. 2019) and PU1K (Qian et al. 2021a), following the official data split for training and testing. PU-GAN dataset consists of 147 objects, with 120 objects split into 24,000 patches for training, and the remaining 27 objects used for testing. PU1K dataset, which includes more data and a wider variety of objects, consists of 1,147 objects. Of these, 1,020 objects are divided into 69,000 patches for training, while the remaining 127 objects are used for testing. Additionally, to evaluate performance on real-scan datasets, we also conducted experiments using the KITTI dataset (Geiger et al. 2013) and the ScanObjectNN dataset (Uy et al. 2019).

Implementation Details. For our experiments, we used an NVIDIA RTX A6000 GPU. The training process followed a curriculum learning strategy, with 50 epochs for easy samples and 50 epochs for hard samples, totaling 100 epochs, with a batch size of 256. We employed the Adam optimizer during training, with an initial learning rate of 0.001. Additionally, we applied random rotation to augment the training samples.

Metrics and Baselines. To evaluate point cloud upsampling performance, we use Chamfer distance (CD), Haus-

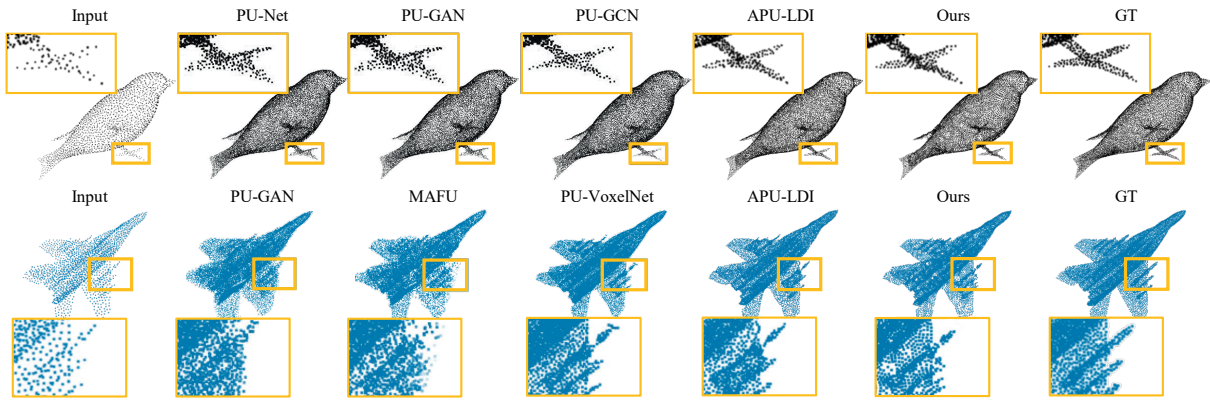


Figure 7: Visualization of upsampling results using synthetic datasets. The top row shows the results on the PU-GAN dataset, while the bottom row presents the results on the PU1K dataset.

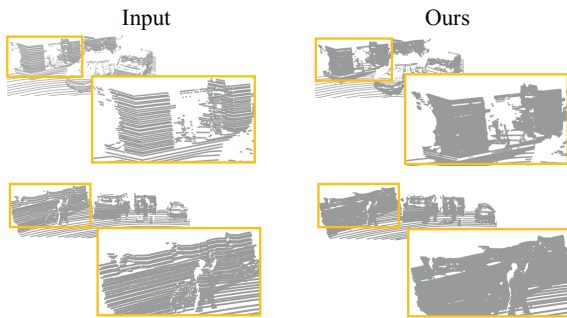


Figure 8: Visualization of upsampling results using KITTI dataset. Objects such as buildings, vehicles, and humans, which are sparsely scanned by the LiDAR sensor, are densely upsampled.

dorff distance (HD), and point-to-surface distance (P2F). The units of CD, HD, and P2F are all 10^{-3} . We make comparison with various traditional and recent state-of-the-art point cloud upsampling methods, including PU-Net (Yu et al. 2018), MPU (Yifan et al. 2019), PU-GAN (Li et al. 2019), Dis-PU (Li et al. 2021), PU-GCN (Qian et al. 2021a), NePs (Feng et al. 2022), Grad-PU (He et al. 2023), APU-LDI (Li et al. 2024), PU-VoxelNet (Du et al. 2024), and RepKPU (Rong et al. 2024).

Results on Synthetic Datasets

PU-GAN Dataset. We conducted experiments on the PU-GAN dataset with upsampling rates of 4 and 16. Table 1 shows that our PLATYPUS outperforms other methods across various metrics at both upsampling rates. Additionally, the top row of Fig. 7 presents the upsampling results of PLATYPUS compared to other methods on the PU-GAN dataset. Notably, in the bird’s foot area, PLATYPUS produces fewer outliers and preserves the silhouette of the foot compared to other methods.

PU1K Dataset. The bottom row of Fig. 7 shows our results on the PU1K dataset. When examining the upsampling

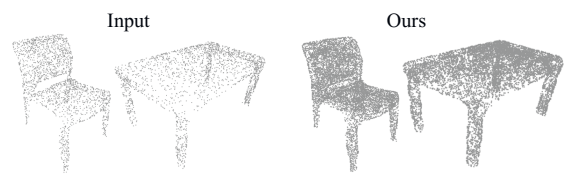


Figure 9: Visualization of upsampling results using ScanObjectNN dataset. Both the real-scanned chair and desk are densely upsampled.

results for the intricate structures on the wings of the jet, it is evident that PLATYPUS preserves the structure and arrangement of the missiles compared to other methods.

Results on Real-Scanned Datasets

To demonstrate that our model, trained on synthetic datasets, generalizes well to diverse real-world scenarios, we conducted upsampling experiments on the outdoor real-scanned KITTI dataset (Geiger et al. 2013) and the indoor real-scanned ScanObjectNN dataset (Uy et al. 2019).

KITTI Dataset. KITTI dataset is related to traffic scenarios used in autonomous driving and robotics and includes point clouds scanned with LiDAR sensor. We selected point clouds from two different scenes and conducted upsampling experiments. As shown in Fig. 8, our results demonstrate that the our work, PLATYPUS, is well-suited for handling point clouds in real-world scenarios, proving to be a practical and valuable technology.

ScanObjectNN Dataset. ScanObjectNN dataset consists of point clouds scanned from various types of objects. We selected point clouds of chairs and tables to verify whether our method could effectively upsample everyday objects. As shown in Fig. 9, PLATYPUS performed well on real indoor data, demonstrating its effectiveness in these scenarios as well.

Method	5×			7×			13×		
	CD↓	HD↓	P2F↓	CD↓	HD↓	P2F↓	CD↓	HD↓	P2F↓
MAFU (Qian et al. 2021b)	0.252	2.308	1.949	0.207	2.593	2.013	0.177	3.212	2.014
Grad-PU (He et al. 2023)	0.244	2.447	2.459	0.223	2.654	3.325	0.242	4.144	5.703
APU-LDI (Li et al. 2024)	0.193	1.706	1.445	0.148	1.534	1.468	0.134	1.878	1.524
PLATYPUS	0.191	1.327	1.845	0.137	1.433	1.866	0.154	1.494	1.379

Table 2: Quantitative comparisons of arbitrary-scale upsampling performance. The results of upsampling by factors of 5×, 7×, and 13× were compared with those of other methods.

Method	$\tau = 0.01$			$\tau = 0.02$		
	CD↓	HD↓	P2F↓	CD↓	HD↓	P2F↓
PU-Net	0.588	6.182	9.842	1.057	9.954	16.282
PU-GAN	0.435	7.848	7.300	0.815	9.450	14.246
Dis-PU	0.430	6.580	6.954	0.776	8.861	13.934
PU-GCN	0.411	5.001	6.963	0.781	8.926	13.730
Grad-PU	0.423	4.307	6.403	0.730	6.993	11.481
APU-LDI	0.339	3.089	5.167	0.622	6.485	10.984
PLATYPUS	0.368	2.224	6.804	0.622	5.504	10.633

Table 3: Quantitative comparison against other methods for two Gaussian noise levels. τ represents the noise level (standard deviation) of the Gaussian noise.

Additional Analyses

Arbitrary-Scale Upsampling. Most upsampling methods are constrained to the upsampling rate used during training, such as 4×, meaning they can only upsample by 4 times during inference as well. Even when performing inference with a 16× upsampling rate, it typically involves applying the 4× upsampling process twice. To demonstrate that our method can perform arbitrary-scale upsampling, not just upsampling at a fixed rate (4×), we conducted experiments using a model trained on the PU-GAN dataset with a 4× upsampling rate and applied this model to upsample at rates of 5×, 7×, and 13×. As shown in Table 2, when compared with other methods capable of arbitrary-scale upsampling, PLATYPUS demonstrates excellent performance.

Robustness on Additive Noise. To evaluate whether our model can robustly upsample the correct structure even when the input point cloud’s structure is distorted, we conducted experiments by adding Gaussian noise to the xyz coordinates of the input point cloud from PU-GAN test dataset and then upsampling the perturbed point cloud. Table 3 presents the results of upsampling with various methods, including ours, using the same noisy input. For both noise levels (i.e., standard deviations), 0.01 and 0.02, PLATYPUS demonstrated superior upsampling performance compared to other methods, indicating that the PLATYPUS upsampling system is robust to variations in the input.

Ablation Study. We conducted experiments to evaluate the effectiveness of various techniques used in PLATYPUS.

Curvature-based Sampling	Curriculum Learning	CD↓	HD↓	P2F↓
-	-	0.258	2.720	2.013
✓	-	0.235	1.739	1.983
-	✓	0.249	2.521	1.948
✓	✓	0.229	1.426	1.908

Table 4: An ablation study demonstrates the effectiveness of our proposed curvature-based sampling and curriculum learning strategy.

For the ablation studies, we used the PU-GAN dataset. Table 4 shows the performance changes in PLATYPUS when the core techniques, curvature-based sampling and curriculum learning based on the skewness of the curvature distribution, are included or excluded. When curvature-based sampling is not used, the sparse point cloud bypasses any sampling process, with the entire original point cloud being fed directly into the feature encoder, followed by the distance estimator, which outputs the unsigned distance d . In the absence of the curriculum learning strategy, the training data is not divided into easy and hard samples; instead, all training data is used in every epoch. As shown in Table 4, both curvature-based sampling and curriculum learning significantly improve performance compared to the base setting. Moreover, the best performance is achieved when both techniques are applied together.

Conclusion

In this study, we introduce PLATYPUS, a novel upsampling system that addresses the challenges of outliers and the difficulty of upsampling complex regions in point clouds. Our network, Progressive Local Surface Estimator (PLSE), utilizing a newly proposed curvature-based sampling method, effectively captures local features from intricate areas with high curvature in sparse point clouds. Additionally, the adoption of a curriculum learning strategy allows the network to progressively learn more complex features, leading to better overall performance. While there is room for optimizing memory usage, our future work will focus on improving the efficiency of storing sampled point clouds. Overall, PLATYPUS significantly advances the state-of-the-art in point cloud upsampling, providing a solid foundation for further research and development in this area.

Acknowledgments

This work was supported in part by the IITP RS-2024-00457882 (AI Research Hub Project), NRF RS-2024-00345806, and IITP 2020-0-01361 (AI Graduate School Program at Yonsei University) funded by Korean Government (MSIT).

References

- Alexa, M.; Behr, J.; Cohen-Or, D.; Fleishman, S.; Levin, D.; and Silva, C. T. 2003. Computing and rendering point set surfaces. *IEEE Transactions on visualization and computer graphics*, 9(1): 3–15.
- Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.
- Chibane, J.; Pons-Moll, G.; et al. 2020. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems*, 33: 21638–21652.
- Du, H.; Yan, X.; Wang, J.; Xie, D.; and Pu, S. 2024. Arbitrary-Scale Point Cloud Upsampling by Voxel-Based Network with Latent Geometric-Consistent Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 1626–1634.
- Duan, Y.; Zhu, H.; Wang, H.; Yi, L.; Nevatia, R.; and Guibas, L. J. 2020. Curriculum deepsdf. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, 51–67. Springer.
- Feng, W.; Li, J.; Cai, H.; Luo, X.; and Zhang, J. 2022. Neural points: Point cloud representation with neural fields for arbitrary upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18633–18642.
- Foorginejad, A.; and Khalili, K. 2014. Umbrella curvature: a new curvature estimation method for point clouds. *Procedia Technology*, 12: 347–352.
- Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.
- He, Y.; Tang, D.; Zhang, Y.; Xue, X.; and Fu, Y. 2023. Gradpu: Arbitrary-scale point cloud upsampling via gradient descent with learned distance functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5354–5363.
- Huang, H.; Li, D.; Zhang, H.; Ascher, U.; and Cohen-Or, D. 2009. Consolidation of unorganized point clouds for surface reconstruction. *ACM transactions on graphics (TOG)*, 28(5): 1–7.
- Li, R.; Li, X.; Fu, C.-W.; Cohen-Or, D.; and Heng, P.-A. 2019. Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7203–7212.
- Li, R.; Li, X.; Heng, P.-A.; and Fu, C.-W. 2021. Point cloud upsampling via disentangled refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 344–353.
- Li, S.; Zhou, J.; Ma, B.; Liu, Y.-S.; and Han, Z. 2024. Learning continuous implicit field with local distance indicator for arbitrary-scale point cloud upsampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3181–3189.
- Lipman, Y.; Cohen-Or, D.; Levin, D.; and Tal-Ezer, H. 2007. Parameterization-free projection for geometry reconstruction. *ACM Transactions on Graphics (ToG)*, 26(3): 22–es.
- Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 165–174.
- Qian, G.; Abualshour, A.; Li, G.; Thabet, A.; and Ghanem, B. 2021a. Pu-gcn: Point cloud upsampling using graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11683–11692.
- Qian, Y.; Hou, J.; Kwong, S.; and He, Y. 2020. PUGeoNet: A geometry-centric network for 3D point cloud upsampling. In *European conference on computer vision*, 752–769. Springer.
- Qian, Y.; Hou, J.; Kwong, S.; and He, Y. 2021b. Deep magnification-flexible upsampling over 3d point clouds. *IEEE Transactions on Image Processing*, 30: 8354–8367.
- Rong, Y.; Zhou, H.; Xia, K.; Mei, C.; Wang, J.; and Lu, T. 2024. RepKPU: Point Cloud Upsampling with Kernel Point Representation and Deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21050–21060.
- Uy, M. A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1588–1597.
- Wu, S.; Huang, H.; Gong, M.; Zwicker, M.; and Cohen-Or, D. 2015. Deep points consolidation. *ACM Transactions on Graphics (ToG)*, 34(6): 1–13.
- Yifan, W.; Wu, S.; Huang, H.; Cohen-Or, D.; and Sorkine-Hornung, O. 2019. Patch-based progressive 3d point set upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5958–5967.
- Yu, L.; Li, X.; Fu, C.-W.; Cohen-Or, D.; and Heng, P.-A. 2018. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2790–2799.
- Zhao, W.; Liu, X.; Zhai, D.; Jiang, J.; and Ji, X. 2023. Self-supervised arbitrary-scale implicit point clouds upsampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10): 12394–12407.
- Zhao, W.; Liu, X.; Zhong, Z.; Jiang, J.; Gao, W.; Li, G.; and Ji, X. 2022. Self-supervised arbitrary-scale point clouds upsampling via implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1999–2007.

Zhou, Q.-Y.; Park, J.; and Koltun, V. 2018. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*.