석 사 학 위 논 문
Master's Thesis

3D 표현 학습을 위한 점진적 곡률 인코딩

Progressive Curvature Encoding
for 3D representation learning

2023

김 소 희 (金 昭 喜 Kim, Sohee)

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

석 사 학 위 논 문

3D 표현 학습을 위한 점진적 곡률 인코딩

2023

김 소 희

한 국 과 학 기 술 원

김재철AI대학원

3D 표현 학습을 위한 점진적 곡률 인코딩

김 소 희

위 논문은 한국과학기술원 석사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2023년 12월 18일

심사위원장 양 은 호 (인)

심 사 위 원 신 진 우 (인)

심 사 위 원 황 성 주 (인)

Progressive Curvature Encoding for 3D representation learning

Sohee Kim

Advisor: Eunho Yang

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Master of Science in AI

Daejeon, Korea
December 18, 2023

Approved by

Eunho Yang
Professor of Kim Jaechul Graduate School of AI

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

MAI 김소희. 3D 표현 학습을 위한 점진적 곡률 인코딩. 김재철AI대학원 . 2023년. 22+iv 쪽. 지도교수: 양은호. (영문 논문)
Sohee Kim. Progressive Curvature Encoding for 3D representation learning. Kim Jaechul Graduate School of AI . 2023. 22+iv pages. Advisor: Eunho Yang. (Text in English)

초 록

3D 포인트 집합을 이해하는데 있어서 곡률 정보는 복잡한 기하학적 패턴을 식별하고 구별하는데 중요한 역할을 한다. 그러나 그 중요성에도 불구하고 대부분의 포인트 클라우드 인코더는 형태 정보를 추출하기 위해 유클리드 좌표 특징에만 의존한다. 일부 선행 연구들이 곡률 정보를 포인트 피쳐에 담고자 하는 시도들을 하였지만, 국소 부위만 보아 더 넓은 시각의 곡률 정보를 이해하지 못한다. 또한 방향에 따라 3차원 물체의 윤곽 정보가 달라짐에도 각 방향에 따른 곡률 정보를 고려한 방법은 아직 제시된 바가 없다. 이런 문제들에 대응하기 위해, 본 논문은 방향 종속적인 다중 계층 곡률 특징을 점진적으로 모델링하는 효과적인 방법을 제안한다. 구 프레임 내의 특정 구간에 속하는 법선 벡터의 차이를 관찰하여 지역 곡률을 추정하고 포인트 클라우드를 평면에 투영하여 얻은 여러 방향의 깊이 맵의 안내로 전역 윤곽 특징을 배우도록 한다.

핵 심 낱 말 기계 학습, 딥러닝, 3차원 표현 학습, 포인트 클라우드 표현 학습, 포인트 클라우드 분석

Abstract

Understanding unordered 3D point sets is crucial for identifying and differentiating complex geometric patterns, and curvature features play a pivotal role in this process. Despite their significance, many point cloud encoders predominantly rely on Euclidean coordinate features for shape information extraction. While some prior works have proposed incorporating curvature information into point features, they often lack a comprehensive consideration of coarse-grained curvature from a broader perspective, focusing solely on local curvature based on nearest points. This limitation becomes more evident in real-world point clouds with high irregularities and local noise. Moreover, existing methods often overlook curvature along different directions, despite the fact that the contour of 3D objects can vary based on their orientation. To address these challenges, this thesis introduces an effective curvature encoding strategy named Progressive Curvature Scanning (**PCS**). This approach models direction-dependent curvature features at multiple levels of granularity, estimating local curvature within the frame of a sphere by analyzing differences in normal vectors belonging to specific spherical sections. Additionally, global contour features are encoded with the assistance of multi-view 2D depth maps obtained through the projection of point clouds.

Keywords Machine Learning, Deep Learning, 3D representation learning, Point Cloud representation learning, Point Cloud analysis

Contents

Contents i
List of Tables iii
List of Figures iv

Chapter 1. Introduction **1**

Chapter 2. Analysis of Curvature Features **3**
2.0.1 Preliminaries and Notations 3
2.0.2 Preliminary Studies for our Hypothesis 3

Chapter 3. Progressive Curvature Scanning **6**
3.0.1 Multi-Directional Local Curvature (MDLC) 7
3.0.2 Multi-View Global Contour (MVGC) 8

Chapter 4. Experiments **10**
4.0.1 Implementation Details 10
4.0.2 3D Semantic Segmentation 10
4.0.3 3D Object Part Segmentation 11
4.0.4 3D Object Classification 11
4.0.5 Domain Shift Conditions 12
4.0.6 Ablation Study 12

Chapter 5. Related Work **13**

Chapter 6. Conclusion **14**

Chapter 7. Appendix **15**
7.1 Additional Experimental Results 15
7.1.1 Semantic Segmentation on S3DIS 15
7.2 Qualitative Analysis 15
7.3 Detailed Experiment Setup 15
7.3.1 Dataset Statistics of PointDA-10 dataset 15
7.3.2 Implementation Details 17
7.3.3 Evaluation Protocol 17

Bibliography **18**

Acknowledgments **21**

Curriculum Vitae **22**

List of Tables

4.1 3D segmentation result on S3DIS (Area-5) 10
4.2 3D part segmentation result in ShapeNetPart 11
4.3 3D object classification result in ModelNet40 11
4.4 3D object classification result in PointDA-10 12
4.5 Ablation study of PCS 12

7.1 3D segmentation result on S3DIS (6-fold) 15
7.2 Data statistics for the PointDA-10 dataset 16

List of Figures

2.1 Simple curvature encoding baselines for the Preliminary Study 4
2.2 Result of the Preliminary Study 4

3.1 Overview of PCS 6

4.1 Visualization of Part Segmentation 11

7.1 Visualization of S3DIS Area-5 segmentation 16

Chapter 1. Introduction

Point cloud analysis has garnered considerable attention in 3D understanding, due to its usage as the primary data format obtained by scanning 3D scenes via depth-sensing cameras (*e.g.* LiDAR sensors). In recent years, deep neural networks have shown promising results across a wide range of point cloud understanding tasks, leveraging convolution [1, 2], message passing [3] and self-attention [4, 5] as geometric pattern extractors. While designs of deep encoders have become increasingly sophisticated, the 'point attribute' itself has not received the same level of attention. Most models rely on 3D coordinates as point features to capture geometric properties, despite their irregular nature and susceptibility to noise. Thus, it is essential to devise elaborate features that comprehensively describe 3D data, which can not only improve performance in parameter-efficient networks but also enhance the performance of large-scale architectures.

To better express the 3D structure, multiple works have been proposed for designing point attributes in addition to Euclidean coordinates. These methods include angle-based relational feature [6], mapping of coordinates into a canonical pose [7, 8] to achieve rotation invariance, and the use of (un)signed distance field for continuous feature encoding [9, 10]. An emerging direction in this field is the incorporation of *curvature information* into point features [11, 12]. Prominent theories in neuroscience field [13, 14, 15, 16] suggest that neural signals derived from curvilinear features facilitate faster and easier detection of 3D visual concepts than those from rectilinear features. In a similar fashion to its importance in human visual perception, the curvature information inherent in 3D-structured data also plays a vital role in deep representation learning for various 3D vision tasks (*e.g.* local minima/maxima of curvatures could represent distinctive regions of 3D data). Despite its importance, this aspect of information has not yet been adequately explored in the context of the point features.

To address this issue, several approaches for estimating curvature have been introduced to integrate it into point features. These techniques conventionally rely on face normal vectors or covariance matrices [17] acquired from nearby points. A recent pioneering work, RepSurf [11], identifies local curvature for each point based on normal vectors of adjacent triangles, which are constructed from surrounding neighboring points in an umbrella shape. While RepSurf has demonstrated remarkable performances on 3D vision task, it still falls short in capturing *coarse-grained* curvature information in a wider context due to its sole focus on local regions. The ability to capture global curvature knowledge becomes critical in natural world scenarios where point clouds exhibit irregularities such as sparse density and occlusion. In these realistic conditions, coarse-grained curvature information can easily provide more accurate detection of a 3D shape, while local curvature features may introduce noisy supervisory signals to models. Another potential limitation of existing methods is the absence of consideration for curvature differences in each *direction*. The curvature of an object typically varies depending on its orientation and scale and aggregating it into a single feature sacrifices a significant amount of shape information. Therefore, encoding multi-directional curvature is also a crucial factor toward precisely understanding the 3D curves of objects.

Motivated by prior insights, we propose **P**rogressive **C**urvature **S**canning (**PCS**), a novel curvature encoding approach for 3D point clouds that can identify directional contour information in a multi-resolution perspective. Our method considers two different scales of curvature features - local curvature and global shape features. The local curvature feature is estimated in a direction-wise manner under an

equally divided sphere centered on each point. Curvature for each direction is determined by the angle differences between the normal vectors of surfaces constructed via points belonging to the corresponding partition in a sphere. Meanwhile, the feature of global curvature (*i.e.* coarse-grained curvature) is efficiently derived with the aid of multi-view 2D depth maps, which possess geometrical contour information. Toward this, the farthest point set of a 3D object is projected onto pre-defined planes while pixel values represent depth along each orthogonal axis. It is worth noting that the receptive fields of our curvature features are gradually adjusted according to those of the encoder blocks, which enables our method to be architecture-agnostic. By doing so, we can *progressively* capture curvature information, viewing from local to global shape aligned with each block's receptive field.

In summary, our contribution is three-fold:

- We investigate the impact of encoding multi-resolution curvature features with the guidance of directional information on the model's comprehension of 3D shape knowledge.
- We introduce local and global curvature as a guide for 3D representation learning. We intend to open the view for utilizing multi-resolutional and multi-directional curvature information for deep representation learning in 3D point cloud data.
- Through an extensive set of experiments, we demonstrate that our method, PCS, yields substantial improvement in a variety of 3D recognition tasks, including scene segmentation, part segmentation, object classification, and its domain shift variants.

Chapter 2. Analysis of Curvature Features: Resolution and Direction Perspectives

We conduct a preliminary study to verify our two key research hypotheses: the importance of encoding multi-*resolution* and *directional* curvature into features. Prior to delving into the studies, we provide a brief outline of our problem settings.

2.0.1 Preliminaries and Notations

Let $\boldsymbol{P} \in \mathbb{R}^{m\times 3}$ be the point cloud data represented as an unordered set of 3D coordinates $\boldsymbol{P} = \{\mathbf{x}_1, \dots \mathbf{x}_m\}$, where m is the number of points and $\mathbf{x}_i \in \mathbb{R}^3$ is a coordinate of i-th point. In this work, without loss of generality, we consider the model consisting of L *set abstraction* layers as a point cloud encoder, similar to representative architectures such as PointNet++ [18], PV-RCNN [19], and PointNeXt [20]. To hierarchically extract features from point clouds, a set abstraction layer comprises three sub-layers: a subsampling layer that downsamples the incoming points using farthest point sampling (FPS), a grouping layer that queries and groups' nearest neighbors for each point, and lastly, multi-layer perceptrons (MLPs) with a reduction operation that are applied to aggregate features of the neighboring points. In summary, the set abstraction process of point $\mathbf{x}_i$ with its neighbor points $\mathcal{N}_i$ is formulated as:

$$\mathbf{z}_i^{l+1} = \psi\big\{h_\theta(\mathbf{z}_j^l, \mathbf{x}_j - \mathbf{x}_i) \mid \mathbf{x}_j \in \mathcal{N}_i\big\}, \tag{2.1}$$

where $\mathbf{z}_j^l$ denotes the input feature of neighbor point x_j at the l-th layer, ψ is the reduction operation (*e.g.* max-pooling), and h_θ represents the MLPs that take the relative vector $(\mathbf{x_j} - \mathbf{x_i})$ and $\mathbf{z}_j^l$ as input. We now investigate which types of curvature features facilitate the model to properly understand and learn the 3D shape information within this encoder setting.

2.0.2 Preliminary Studies for our Hypothesis

We scrutinize the efficacy of various curvature encoding strategies, with a focus on 'scale' and 'direction' perspectives of curvature information. To validate these strategies in realistic scenarios, we train the models equipped with different curvature features and evaluate their performance under diverse *distributional shift* conditions. In such conditions, curvature features play a more crucial role than coordinate features as deep models encounter point clouds that exhibit distributional shift caused by variations in object scales, point density, and object occlusion. Toward this, we devise multiple curvature encoding baselines and compare them using 3D shape classification benchmark dataset, PointDA-10 [21], which is constructed by extracting data in 10 shared classes from three datasets - ModelNet40 [22], ShapeNet [23] and ScanNet [24].

Multi-resolution curvature We design three simple baselines considering global curvature to support our assumption about the advantage of capturing curvature information across a wider range (*i.e.* coarse-grained curvatures). The comparison is made between our baselines and a local curvature encoding approach, which is analogous to the prior work [11]. For *Local curvature* baseline (Figure 2.1 (a)), we sample k nearest neighbor points and construct k adjacent triangle in a counter-clockwise direction. Subsequently, we derive normal vectors for each triangle surface and combine them with the input

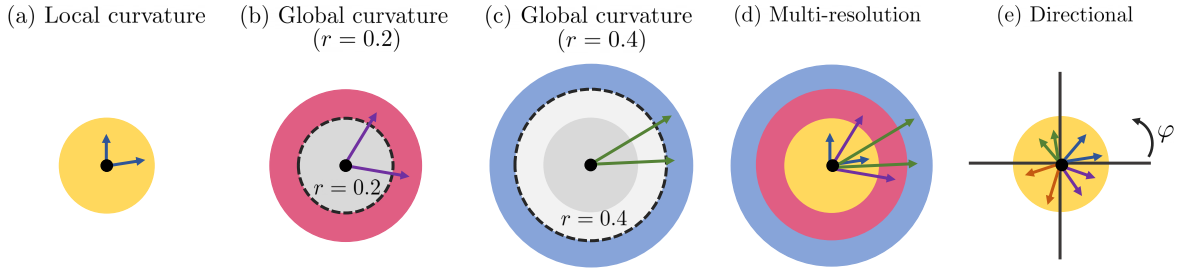

Figure 2.1: Simple curvature encoding baselines for conducting experiment to support our assumption. Each neighbor sampling method varies by : (a) k-NN, (b) k-NN over radius 0.2, (c) k-NN over radius 0.4, (d) average of (a)-(c), (e) sampling points from explicit directions.

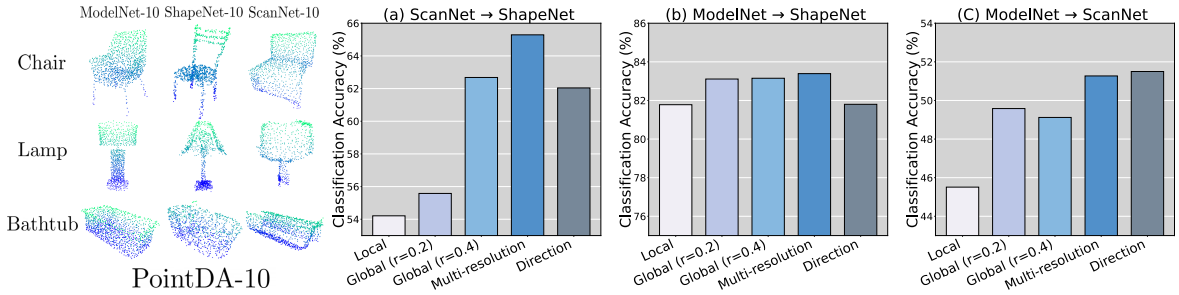

Figure 2.2: The result of our analysis on curvature features. As shown in the most left figure, each domain has completely different shape characteristics. In order to perform well on these shifts, it is important to learn more general information, and entire domain transition results show that our proposed multi-resolution and multi-directional features outperform the single local curvature feature.

coordinate features using a linear layer. In contrast, *Global curvature* baselines (Figure 2.1 (b) and (c)) similarly construct curvature features but restrict the selection of neighboring points *over* a certain radius r, enforcing the exploration of broader regions. We set $r \in \{0.2, 0.4\}$ and $k = 8$, respectively. Lastly, *Multi-resolution curvature* (Figure 2.1 (d)) is obtained by jointly considering the normal vectors from different scales. Specifically, a set of normal vectors for multi-resolution curvature features are computed by simply averaging three sets of normal vectors acquired at three different scales: $\{\mathbf{n}_i^{\text{multi}}\}_{i=1}^k = \{(\mathbf{n}_i^{r_0} + \mathbf{n}_i^{r_{0.2}} + \mathbf{n}_i^{r_{0.4}})/3\}_{i=1}^k$, where $\mathbf{n}_i^{r_0}$, $\mathbf{n}_i^{r_{0.2}}$, and $\mathbf{n}_i^{r_{0.4}}$ denote the i-th normal vector with radius $r = 0$ (local curvature), $r = 0.2$, and $r = 0.4$. Here, averaging makes sense since normal vectors are sorted counter-clockwise starting from a fixed reference axis (*i.e.* positive x-axis), making normal vectors within a similar circular sector being averaged.

Using the described curvature features, we train PointNet++ [18], a representative point cloud encoding architecture, and evaluate models over three different distributional shift settings in PointDA-10 dataset (Figure 2.2). Compared with the local curvature feature, we confirm that injecting more coarse-grained curvature features in the point cloud encoding process allows the model to learn versatile shape knowledge that is applicable across domains. In particular, leveraging *multi-resolution* curvature information consistently exhibits superiority over other curvature features implying the necessity of jointly encoding multi-scale curvatures into point features.

Directional curvature We also explore the effectiveness of incorporating direction information into curvature features (informally speaking, we let the model be aware of curvature for each direction). To

this end, we design *Directional curvature* encoding approach (Figure 2.1 (e)) that estimates direction-wise curvature within a sphere. The sphere, centered on each seed point, is evenly divided into n partitions using two angles of the spherical coordinate system: the polar angle θ, which measures the angle with respect to the positive z-axis, and azimuthal angle φ, which is the angle of the orthogonal projection of a vector onto the xy-plane relative to a reference direction, *i.e.* positive x-axis. Within each partition of the sphere, we sample the nearest points and calculate the normal vectors by constructing surfaces using the sampled points. To give supervision about the direction of obtained normal vectors, we set the average of sampled point coordinates as a direction indicator. In other words, the normal vectors for each partition are used as curvature features along with their corresponding direction coordinate. This directional curvature encoding differs from the local curvature baseline in that simple k-nearest neighbors (k-NN) sampling results in neighbors with arbitrary orientations which can be sensitive to local noise, and it is challenging to capture the direction-wise curvature pattern. For our experiments, we set $n = 8$ and ensure the number of parameters in the linear layers are matched for fair comparison with other curvature feature types.

We evaluate the efficacy of the directional curvature feature using the same experimental setup as described in the previous paragraph. As shown in Figure 2.2, the directional curvature features outperforms k-NN based local curvature features across multiple domain shift scenarios. These observations suggest that directional curvature knowledge enhances the model's ability to universally understand 3D shapes.

Chapter 3. Progressive Curvature Scanning

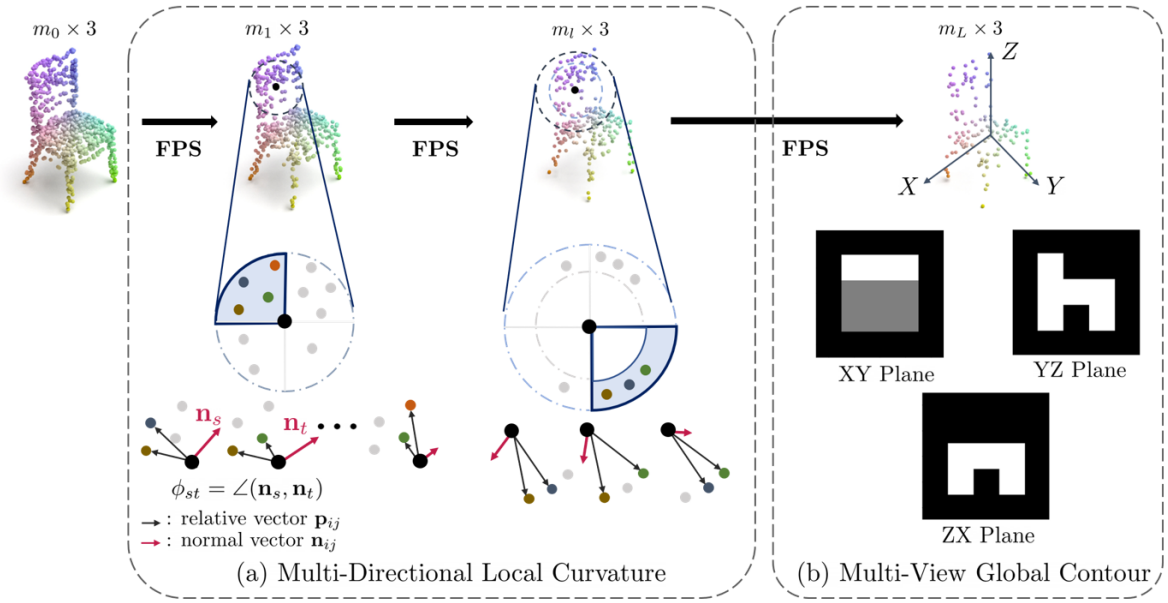

Figure 3.1: An overview of our feature encoding method, Progressive Curvature Scanning(PCS). Given a point cloud with m_0 points, as it goes through multiple sub-sampling stages, we compute multi-directional (multi-view) curvature feature in a gradual manner. We propose two types of curvature, (a) a local curvature, MDLC, and (b) a global curvature, MVGC.

As demonstrated in Section 2, conjugating multi-resolution silhouettes with direction knowledge is essential for grasping the underlying shapes of irregular and noisy point clouds, as well as for learning general shape representations. Inspired by these in-depth analyses, we introduce Progressive Curvature Scanning (PCS), a tailored solution for modeling curvature features and blending them into the point cloud encoding process. Our method consists of two distinct scales of curvature features: Multi-directional Local Curvature (MDLC) and Multi-view Global Contour (MVGC). Under the sphere frame, MDLC evaluates local curvature in a direction-wise manner using differences of normal vectors (Section 3.0.1). Meanwhile, for capturing global contour information, we introduce MVGC in Section 3.0.2, which extracts depth information in the perspective of multi-angles by projecting point clouds into pre-defined planes, *i.e.* 2D depth-maps. Note that local curvature features (MDLC) are adaptively computed for each model block through a varying radius of sphere aligned with the receptive field of the model block. As for the last block, global contour features (MVGC) are injected. The intuition behind this is that, by doing so, curvature features can be *progressively* captured from fine-grained to coarse-grained scale while associating with the receptive field of existing point cloud encoders. We describe the details of each component in the following subsections. The overall procedure of PCS is presented in Figure 3.1.

3.0.1 Multi-Directional Local Curvature (MDLC)

In this subsection, we propose a new sampling and computing strategy for local curvature. In the field of point cloud analysis, neighbor sampling process such as k-NN or sampling within a radius boundary (*i.e.* ball query) are often used. Previous works on curvature encoding has also predominantly relied on k-NN, which solely depends on distance between points, making it susceptible to variations of point density. We propose a *multi-directional* sampling method, which considers both distance and angle by sampling uniformly from all directions within a sphere, and further establish a radius lower bound to sample non-overlapping points for each hierarchical block. Note that this method is only applied for MDLC learning and the original model block's grouping continues to use the original ball-query method described in the original paper [18].

Subdividing local region Here we describe our method via a layer-wise processing step. At each layer l, we pass m_{l-1} input points from the previous layer $l-1$, and subsample m_l centroids $\boldsymbol{P}^{(l)} \in \mathbb{R}^{m_l \times 3}$ through farthest point sampling (FPS). For each centroid $\mathbf{x}_c$, input points within the sphere of radius r_l become neighbor candidates, and we explicitly assign them into equally divided n regions based on the angles of spherical coordinates: polar angle θ, which measures the angle with a fixed z-direction, and azimuthal angle φ, which is the angle of the orthogonal projection of vector on to the xy plane and a fixed reference direction (*i.e.* positive x-direction). For example, we can divide them into 8 octants by setting $\varphi \in \{0, \pi/4, \pi/2, 3\pi/4\}$ and $\theta \in \{0, \pi/2\}$. Then, from each region, k-nearest points are sampled, resulting in a tensor $\mathcal{N}^{(l)} \in \mathbb{R}^{m_l \times n \times k \times 3}$. As the layer gets deeper, the radius r_l increases and the number of FPS points m_l decreases, so as to progressively abstract broader local regions. However, we found that k-NN points do not vary much with respect to each block, so we set the previous block's radius r_{l-1} as a minimum radius bound, generating a donut-like receptive field. As a result, we can obtain a more evident multi-resolutional curvature.

Defining local curvature Secondly, we explain our definition of local curvature. For a centroid $\mathbf{x}_c$, there exists $n \times k$ neighbor points $\mathcal{N}_c$. However, as we calculate curvature for each region respectively, our method will be explained in a single region's perspective, and we denote the set of points in the region as $\mathcal{R}_g$. First, make k relative vectors $\mathbf{p}_{cj} = \mathbf{x}_j - \mathbf{x}_c$ using $\{\mathbf{x}_j | (\mathbf{x}_j \in \mathcal{N}_c) \cap (\mathbf{x}_j \in \mathcal{R}_g)\}$ and create ${}_kC_2$ relative vector combinations, which will act as individual surfaces. Through cross product of pairwise relative vectors, we obtain normal vectors $\mathbf{N}_c = \{\mathbf{n}_t\}_{t=1}^{{}_kC_2}$. Due to the sign ambiguity of cross products, we compute the inner product of centroid vector $\mathbf{x}_c$ and normal vector $\mathbf{n}_t$, and choose the direction that makes this value positive, inducing the normal vectors to face outward and not towards the global centroid. Then, compute angle differences of each normal vector pair by doing cosine inverse of the cosine similarity.

$$\forall \mathbf{n}_s, \mathbf{n}_t \in \mathbf{N}_c, \quad \phi_{st} = \cos^{-1} \frac{\mathbf{n}_s \cdot \mathbf{n}_t}{\|\mathbf{n}_s\| \|\mathbf{n}_t\|} \tag{3.1}$$

We use average and max angle as the region $\mathcal{R}_g$'s curvature information.

$$\phi_c^{\text{avg}} = \sum_{s=1}^{{}_kC_2-1} \sum_{t=s+1}^{{}_kC_2} \frac{2\phi_{st}}{|\mathbf{N}_c|(|\mathbf{N}_c|-1)}, \quad \phi_c^{\max} = \max(\{\phi_{st} | s \in [1, {}_k C_2 - 1], t \in [s+1, {}_k C_2]\}) \tag{3.2}$$

In addition, we also give the direction information $\mathbf{v}_c$ of the region by averaging relative vectors $\{p_{cj}\}_{j=1}^{k}$, in order to encode orientational curvature. As a result we obtain our MDLC feature, $\mathbf{f}_c^g = \{\phi_c^{\text{avg}}, \phi_c^{\max}, v_c\}$, for each centroid $\mathbf{x}_c$ and region $\mathcal{R}_g$.

There have been previous attempts in encoding surface representation, via predicted normal vector itself or the eigenvalues of a covariance matrix defined by the neighboring points. However, the former relies on a normal vector defined by a single triangle plane, which is not suitable for point clouds that are easily exposed to noise, and the latter only possess the information of how the points are dispersed through each principal axes. In order to overcome these limitations, our method takes advantage of multiple normal vectors and use their relative angle difference as the local geometry feature.

To plug-in our feature into the baseline model, we abstract our feature through multiple 2D convolution layers. We pair features that are in the same θ region and adjacent φ region in a counter-clockwise manner $[f_c^g, f_c^{g'}] \in \mathbf{R}^{5\times 2}$, then pass it through (1,2) convolution layer. Rather than individually encoding multi-regional curvature, we mix the features that are side by side to incorporate multi-directional curvature when extracting feature.

3.0.2 Multi-View Global Contour (MVGC)

We now introduce how our method encodes multi-angular global contour of 3D data into features by leveraging 2D *depth maps*. Although MDLC effectively captures the local curvature of a wider region as layer l increases, it is still limited to viewing the surrounding area centered on a specific point. Different from these local features, our Multi-View Global Contour (MVGC) features are generated by incorporating the depth information from the *entire* range of points. In the following paragraphs, we provide a detailed explanation for acquiring 2D depth maps containing substantial spatial information and use them as a global curvature feature.

Depth map generation To obtain multi-view depth maps from a 3D point cloud, we begin by projecting the farthest point set of last layer L, $\tilde{\boldsymbol{P}}^{(L)}$, onto three orthogonal planes - xy, yz, and zx planes. For brevity, let us focus on depth map for the xy plane. We suppose a set of projected coordinates, denoted as $\tilde{\boldsymbol{P}}^{xy} = \{\mathbf{x}_i^{xy}\}_{i=1}^{m_L}$, where z-axis values are suppressed. Then, to identify the scale of projected point cloud, the maximum length $d_{\max}$ along two axes is computed as: $d_{\max} = \max_{i,j\in\tilde{\boldsymbol{P}}^{xy}} \left\{ \max|\mathbf{x}_{ix} - \mathbf{x}_{jx}|, \max|\mathbf{x}_{iy} - \mathbf{x}_{jy}| \right\}$, where x_{ix} and x_{iy} represent the x-axis value and y-axis value of $\mathbf{x}_i^{xy}$, respectively. Given a predefined image size $W\times W$, each grid size is set as $\frac{d_{\max}}{W} \times \frac{d_{\max}}{W}$, and we assign the projected point to its matching grid. Lastly, pixel value of individual grid is determined by the summation of z-axis values of the points fitting in it, which reflect depth relation between the points. This process allows us to efficiently acquire unified size of depth maps $H \in \mathbb{R}^{W\times W}$ from irregular scale point clouds without relying on any off-the-shelf rendering tools [25, 26] or time-consuming transformations [27]. We define H_{xy}, H_{yz}, H_{zx} as 2D depth map for the xy, yz, zx-planes, respectively.

Encoding global contour information Armed with depth maps from three different planes, denoted as $H_{xy}, H_{yz}, H_{zx} \in \mathbb{R}^{W\times W}$, we extract global contour features by applying 2D convolution operations to them. The global contour feature z_{MVGC} for point cloud $\boldsymbol{P}$ is derived by combining the feature obtained from the different views as follows:

$$z_{\text{MVGC}} = \texttt{Combine}\{\Phi(H_{xy}), \Phi(H_{yz}), \Phi(H_{zx})\}, \tag{3.3}$$

where Φ is the shared convolution layers, and `Combine` operation is simply selected between averaging and one linear projection layer. The resulting global contour features are then concatenated with the output features of the layer $L-1$ and injected into the last layer L, which usually has the largest receptive field. By injecting the multiple-view global contour tendencies into the point cloud encoding path, the

model can learn the global shape prior of 3D data, exhibiting robustness over the noisy nature of points. Overall, our proposed features take the best of both worlds by combining the strengths of global and local curvature features, providing a comprehensive representation of shape that is versatile in a wide range of 3D vision tasks.

Chapter 4. Experiments

We conduct a series of experiments to evaluate our method, Progressive Curvature Scanning (PCS), compared to other feature encoding baselines. First, we test our curvature feature on various 3D vision tasks using standard benchmark datasets, including *S3DIS* for semantic segmentation (Section 4.0.2), *ShapeNetPart* for part segmentation (Section 4.0.3), and *ModelNet40* for object classification (Section 4.0.4). To verify our claim that the model trained with the proposed PCS features exhibits robustness to noise and distributional shift, we also conduct experiments on *PointDA-10* dataset, which has six different domain shift scenarios. Lastly, we conduct an ablation study to validate the significance of each component in our method (Section 4.0.6).

4.0.1 Implementation Details

In all experiments, we use a sphere equally divided into 8 regions (n=8) for the Multi-Directional Local Curvature (MDLC) feature, *i.e.* $\varphi \in \{0, \pi/4, 2\pi/4, 3\pi/4\}$ and $\theta \in \{0, \pi/2\}$. Regarding the Multi-View Global Contour (MVGC) feature, we set the size of the 2D depth map $W \times W$ to be 32×32, except for part segmentation task which uses 16×16. We adopt PointNet++ [18] as the default point cloud encoding architecture, in line with prior works [11, 12]. The detailed evaluation protocol and implementation details of our method are provided in Appendix.

4.0.2 3D Semantic Segmentation

S3DIS [28] S3DIS is a challenging dataset that contains 6 large-scale indoor areas with 13 semantic categories. We report Area-5 result and leave the rest of the results in the Appendix. As each scene varies in its point size, we use the offset instead of batch following [11]. As shown in Table 4.1, our method is on par with or exceeds RepSurf in most of the classes and the overall performances. Especially for 'column' and 'door', which are easy for RepSurf as it encodes the normal vector direction itself, we outperform RepSurf with a meaningful amount.

Table 4.1: **3D segmentation result on S3DIS (Area-5)**. We report mIoU, mAcc, OA, per class IoU. † : used the public result of each paper.

Method	**S3DIS Area-5**															
	mIoU	mAcc.	OA	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet† [29]	41.1	49.0	-	88.8	97.3	69.8	**0.1**	3.9	46.3	10.8	52.6	58.9	40.3	5.9	26.4	33.2
PointNet++† [18]	53.5	-	-	89.4	97.7	75.4	0.0	1.8	58.3	19.5	69.2	79.0	46.2	59.1	58.7	41.6
PnP-3D† [30]	58.7	-	-	90.8	98.1	77.0	0.0	7.6	61.7	28.1	74.2	84.3	67.3	64.8	60.5	48.0
RepSurf [11]	66.8 ± 0.02	74.2 ± 0.01	89.4	**93.6**	97.9	84.7	**0.1**	32.5	60.5	69.5	**78.8**	87.9	70.4	70.0	67.3	55.6
PCS (MDLC only)	68.0 ± 0.02	75.7 ± 0.02	**89.7**	93.1	96.8	**85.8**	0.0	36.2	62.2	**74.0**	76.9	88.4	72.6	**70.9**	**69.6**	**56.7**
PCS (Ours)	**68.3** ± 0.01	**76.6** ± 0.01	89.6	92.9	**98.4**	85.0	0.0	**41.1**	**62.4**	73.8	78.0	**88.8**	**72.8**	70.8	67.6	56.2

4.0.3 3D Object Part Segmentation

ShapeNetPart [31] ShapeNetPart is a widely used dataset for part segmentation in 3D point clouds. It is a synthetic dataset of 16,881 shapes, containing 16 object classes, each composed of 2-5 parts (with 50 part classes in total). We repeat it for 3 times with varying seeds (*i.e.* 2023, 2024, 2025) and report its mean and standard deviation. As shown in Figure 4.1, our method is capable of capturing regions that RepSurf fails due to its dependence on normal vector *orientation* and *single-scale* local feature. Misclassifying the outer part of headset ear and plastic bag (second row) may be caused by adjacent green parts having similar normal vectors, confusing them to learn similar features. Our method sees the *difference* of normal vectors, which is a more intrinsic feature capable of detecting edge parts. Moreover, the existing method is weak at capturing thin strap due to its narrow vision, while our method sees all direction with a broader view, distinguishing it better. The overall performance of ShapeNetPart is reported in Table 4.2.

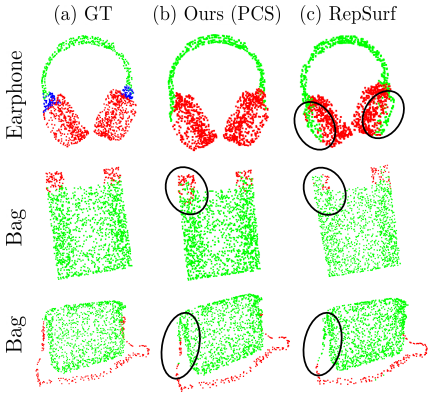

Figure 4.1: **Part Segmentation Visualization.** We visualize GroudTruth, our method and RepSurf's result.

Table 4.2: **3D part segmentation result in ShapeNetPart**. We report the class mIoU and instance mIoU with 3 repetitions.

Method	ShapeNetPart	
	Class mIoU	Instance mIoU
PointNet† [29]	80.4	83.7
DGCNN† [2]	82.3	85.2
PointNet++† [18]	81.58	84.94
GeoSSL† [12]	-	84.8
RepSurf [11]	81.96 ±0.06	84.91 ±0.02
PCS (ours)	**82.36** ±0.06	**85.21** ±0.03

Table 4.3: **3D object classification result in ModelNet40**. We report OA and mAcc. with standard error on 3 repetitions.

Method	ModelNet40	
	OA	mAcc.
PointNet† [29]	89.2	86.0
DGCNN† [2]	92.9	90.2
PointNet++† [18]	90.7	88.4
GeoSSL† [12]	91.7	-
RepSurf [11]	92.67 ±0.08	89.82 ±0.07
PCS (ours)	**93.36** ±0.04	**90.66** ±0.04

4.0.4 3D Object Classification

ModelNet40 [22] ModelNet40 is a synthetic CAD model dataset that contains 9,843 training and 2,468 test data with 40 object categories. It is well known for its hard reproduciblity, so we exclude random drop which increases the randomness, and only use standard scale and shift augmentations. Trained in the exact same setting, our method outperforms RepSurf in a highly saturated dataset 'ModelNet40' by almost 1%, as shown in Table 4.3.

Table 4.4: **PointDA-10 accuracy for 6 settings**. We outperform RepSurf in all settings.

Method	M→S	M→S*	S→M	S→S*	S*→M	S*→S	Avg.
RepSurf	82.62	37.20	77.22	31.66	60.98	57.59	57.88
PCS(Ours)	**83.03**	**52.29**	**80.49**	**47.20**	**62.20**	**59.58**	**64.13**

Table 4.5: **Ablation study of PCS.**

Modules				**ModelNet40**	
MDLC (*single direction*)	MDLC (*single scale*)	MDLC	MVGC	OA	mAcc.
✓				92.75	89.75
	✓			92.79	89.73
		✓		93.04	89.94
		✓	✓	**93.36**	**90.66**

4.0.5 Domain Shift Conditions

PointDA-10 [21] PointDA-10 is constructed by gathering point clouds from 10 shared classes extracted from three distinct benchmark datasets - ModelNet40 [22] (M), ShapeNet [23] (S) and ScanNet [24] (S*). While the point clouds for ModelNet40 and ShapeNet are acquired by sampling points from the surface of 3D CAD models, ScanNet point clouds are collected through scanning real-world scenes. Consequently, distribution shifts are prevalent among these datasets. By Using these sub-datasets, we conduct experiments on six different distributional shift scenarios. As presented in Table 4.4, our method significantly outperforms the most powerful baseline, RepSurf with an average improvement of **6.25%**. These empirical results reinforce our assertion that encoding multi-resolution curvature feature with directional guidance facilitates the learning of comprehensive and versatile knowledge about 3D shapes, leading to enhanced robustness against distributional shifts.

4.0.6 Ablation Study

In Table 4.5, we present an ablation study aimed at verifying the key philosophies in PCS. Firstly, the 'MDLC (*single direction*)' represents our local curvature feature *without* dividing into sphere frames. Neighbor points are sampled without considering their direction. Secondly,the 'MDLC (*single scale*)' involves capturing local curvatures at a single scale, where MDLC features are computed only for the first layer. Lastly, we assess the impact of exclusively exploiting MDLC features for curvature feature encoding, without incorporating the global curvature feature (MVGC). As demonstrated in Table 4.5, the removal of either 'multi-resolution' or 'directional guidance' from the curvature feature results in a decrease in object classification accuracy. Thus, we believe that each module effectively contributes to a comprehensive understanding of 3D shapes.

Chapter 5. Related Work

Local geometry based feature aggregation Local geometrical information is vital in 3D point cloud representation learning, so previous works tried to aggregate neighborhood features using methods from other data types, such as convolution [1, 2] from image and Message Passing [3] from graphs. However, due to the irregularity of point clouds, set abstraction methods [29, 18] still reign this field of work. Rather recently, a new paradigm have been proposed by CurvNet [32], which groups neighbors based on curvature, not distance.

Local feature definition Some work proposed using geometrical information, such as angle [6] and distance [9], as it possess local relational information. Another renowned surface representation is normal vector, and PointNet++ [18] show a performance boost by using this as an external feature. However, in most of the cases this is not given, so a recent work [11] propose predicting normal vector through neighboring points and use it as an additional feature. Our paper lies in the second branch of the research as we propose a new feature that sees multiple direction and scale. A study [33] that came out shortly after ours integrates hierarchical geometry structures, but do not take the directional information into account. Furthermore, we use a global feature, which is not a combination of local features like most of the previous works, that enables the model to see the whole picture at once.

Chapter 6. Conclusion

We have investigated the impact of encoding multi-resolution curvature features, with the guidance of directional information, on the model's comprehension of 3D shape knowledge. Based on our observations, we propose a Progress Curvature Scanning (PCS) strategy, that captures curvature information at a multiple level of granularity. Additionally, our method estimates local curvature using the sphere frame, which facilitates the incorporation of directional information into the curvature features. Through extensive experiments, PCS has demonstrated its effectiveness across various 3D vision recognition tasks even under distributional shift conditions. One potential limitation of our work is that we have not yet explored whether it can be applied to other 3DS structure data such as voxel data and mesh data.

Chapter 7. Appendix

7.1 Additional Experimental Results

7.1.1 Semantic Segmentation on S3DIS

S3DIS 6-fold The S3DIS dataset [28] consists of 271 rooms belonging to 6 large areas. We reveal the quantitative results (mIoU, mAcc, OA) of all 6 areas and compare our method, PCS, with the previous state-of-the-art method RepSurf [11]. As shown in Table 7.1, our method outperforms in Area 1,3, 5 and is on par with RepSurf in Area 4.

Table 7.1: **3D segmentation result on S3DIS (6-fold)**. We report mIoU, mAcc, OA and per class IoU. Both methods (PCS and RepSurf) use 0.98M parameters.

	Method	S3DIS (6-fold)															
		mIoU	mAcc.	OA	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
Area1	RepSurf	75.6	84.7	90.3	96.5	95.8	83.0	69.8	60.7	72.9	87.1	72.5	82.7	65.6	61.6	63.6	70.6
	PCS	**77.4**	**86.3**	**90.7**	96.5	95.9	84.0	69.6	56.6	81.8	85.4	74.2	82.6	73.3	61.8	74.2	70.8
Area2	RepSurf	**58.1**	**72.0**	**86.0**	89.2	96.0	82.5	20.9	50.6	21.3	67.8	54.1	60.6	74.0	53.9	33.8	50.7
	PCS	55.4	69.0	84.3	89.5	95.5	81.9	19.1	49.2	27.1	64.8	56.4	41.4	73.4	49.9	23.8	47.7
Area3	RepSurf	76.8	86.1	91.2	96.1	98.4	83.2	70.7	47.9	51.7	87.3	73.7	82.8	75.7	69.4	88.7	73.0
	PCS	**78.5**	**87.1**	**91.9**	96.0	98.4	84.3	72.7	51.3	53.3	87.0	75.2	84.6	78.4	73.5	90.7	75.1
Area4	RepSurf	**62.7**	**74.7**	**87.8**	93.9	97.8	83.7	13.8	61.2	44.7	68.3	65.6	70.0	55.8	57.6	41.1	61.1
	PCS	62.4	74.1	87.6	94.0	97.9	83.5	17.4	58.0	38.9	65.4	66.9	68.0	54.5	56.6	48.4	62.3
Area5	RepSurf	66.8	74.2	89.4	93.6	97.9	84.7	0.1	32.5	60.5	69.5	78.8	87.9	70.4	70.0	67.3	55.6
	PCS	**68.3**	**76.6**	**89.6**	92.9	98.4	85.0	0.0	41.1	62.4	73.8	78.0	88.8	72.8	70.8	67.6	56.2
Area6	RepSurf	**82.3**	**91.1**	**92.8**	96.4	97.7	86.6	78.3	73.8	81.4	87.7	78.8	85.5	73.9	74.2	81.6	74.2
	PCS	81.5	90.5	92.2	96.4	97.7	85.0	73.7	70.6	82.0	86.6	78.3	84.6	76.0	71.9	83.0	73.3

7.2 Qualitative Analysis

Visualization of segmentation results on S3DIS We visualize several semantic segmentation results from S3DIS Area-5 and compare them with those of the baseline, RepSurf [11]. In Figure 7.1 (a) our model accurately predicts the board (gray) and captures the pink region (column), while RepSurf fails. Comparing two models in (b), our model is less noisy especially when capturing the ceiling (light green) and column (pink). Lastly, in (c), RepSurf incorrectly identifies the ceiling as clutter, while PCS distinguishes it accurately.

7.3 Detailed Experiment Setup

7.3.1 Dataset Statistics of PointDA-10 dataset

As presented in Table 7.2, the PointDA-10 [21] dataset consists of point cloud objects from 10 shared classes obtained from three distinct datasets - ModelNet [22], ShapeNet [23], and ScanNet [24]. Point clouds in both the ModelNet and ShapeNet datasets are collected by sampling points from the surface of 3D CAD models. The ModelNet-10 dataset comprises 4,183 training and 856 test point clouds,

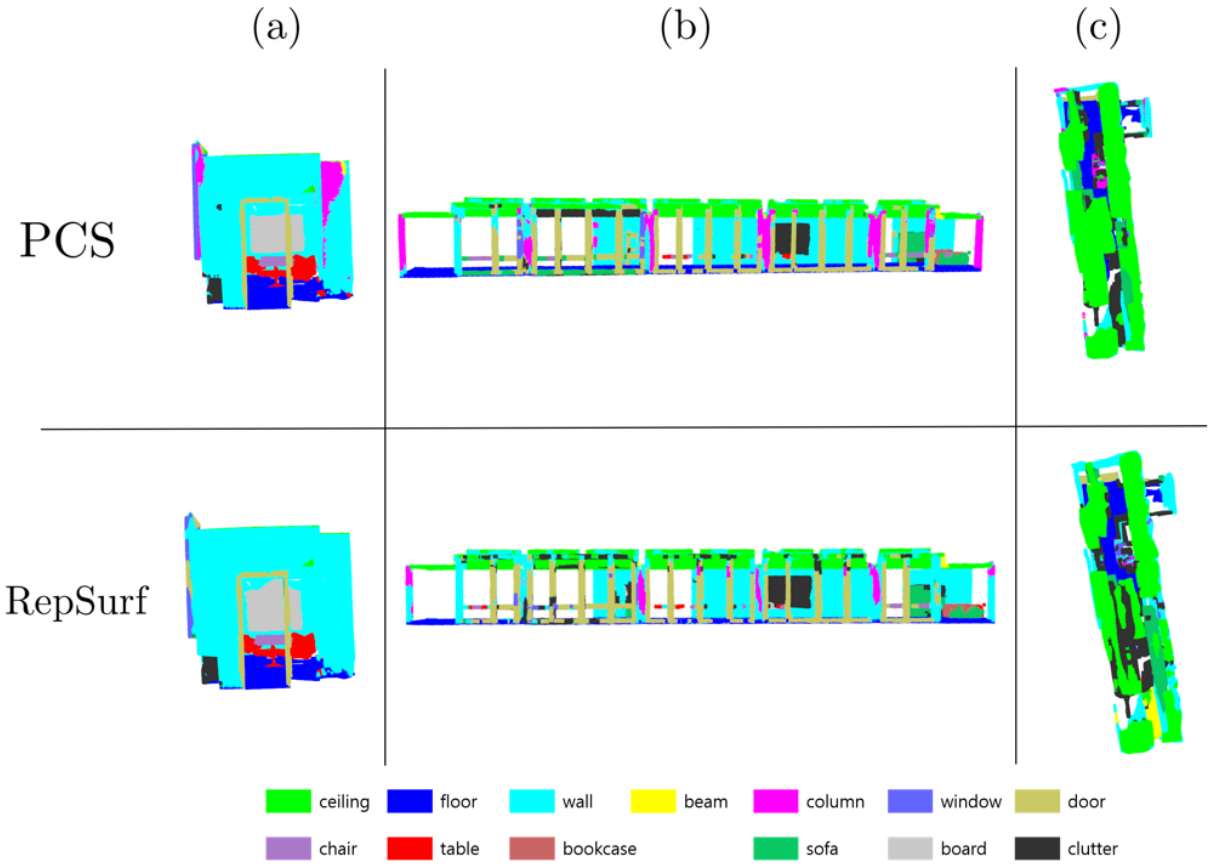

Figure 7.1: **Visualization of S3DIS Area-5 segmentation**: (a) conference room, (b) hallway, and (c) storage.

while ShapeNet-10 contains 17,378 training and 2,492 test point clouds. In contrast, point clouds in the ScanNet are curated through the scanning of real-world scenes. ScanNet-10 includes 6,110 training and 1,769 point clouds with 2048 points each. These point clouds often exhibit partial views of objects due to adjacent objects or self-occlusion. We follow the data pre-processing and data split procedure employed in representative point cloud domain adaptation works [21, 34]. Specifically, the point clouds are aligned with the direction of gravity, while allowing arbitrary rotations along the z-axis. Subsequently, the point clouds are normalized to fit within a unit cube. All point clouds are down-sampled to 1,024 points.

Table 7.2: Data statistics for the PointDA-10 dataset, showing the number of samples per class in three different sub-datasets.

Dataset	**Domain**		bathtub	bed	bookshelf	cabinet	chair	lamp	monitor	plant	sofa	table	**Total**
ModelNet-10	3D CAD	Train	106	515	572	200	889	124	465	240	680	392	4,183
		Test	50	100	100	86	100	20	100	100	100	100	856
ShapeNet-10	3D CAD	Train	599	167	310	1,076	4,612	1,620	762	158	2,198	5,876	17,378
		Test	85	23	50	126	662	232	112	30	330	842	2,492
ScanNet-10	Real	Train	98	329	464	650	2,578	161	210	88	495	1,037	6,110
		Test	26	85	146	149	801	41	61	25	134	301	1,769

7.3.2 Implementation Details

We adopt PointNet++ [18] as the baseline encoder, and plug in our MDLC feature at the end of each block and MVGC at the front of the last block, through concatenation. The MDLC feature is computed using the same seed point but different neighbor points as the set abstraction layer, hence we concatenate it at the end. For segmentation tasks, extra local feature is added at the beginning of each layer to catch a more fine-grained detail. We simply pass relative vectors, made by subtracting seed point from k-NN points at the input layer, through 1-MLP layer and integrate it at the input of each layer. Furthermore, for part segmentation, MDLC feature is also passed to the decoder through skip connection.

7.3.3 Evaluation Protocol

We follow the experimental settings of [11] for the S3DIS dataset [28] and ModelNet40 [22] dataset. For the semantic segmentation task on the S3DIS, we use the AdamW optimizer with a batch size of 8 and a weight decay of 1e-2. The initial learning rate is set to 0.0006 and decreases by a factor of 0.1 at the 60 and 80 epochs, respectively. Following the protocol presented in [11], we apply point cloud augmentations including color contrasting, color shifting, and color jittering. Regarding shape classification on the ModelNet40 dataset, we utilize the Adam optimizer with a batch size of 64, and the weight decay is set to 1e-4. The initial learning rate is set to 0.001 and decays by a factor of 0.7 every 20 epochs. In this experiment, we adopt point cloud scaling and shifting augmentations. Lastly, for part segmentation on the ShapeNetPart [31], we employ the Adam optimizer with a batch size of 16 and a weight decay of 1e-4. The initial learning rate is set to 0.001 and decays by a factor of 0.5 per 20 epoch. The models are trained for 100 epochs on the S3DIS dataset and 250 epochs on both the ModelNet40 and ShapeNetPart datasets.

Bibliography

[1] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8895–8904, 2019.

[2] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[3] Guohao Li, Matthias Müller, Guocheng Qian, Itzel Carolina Delgadillo Perez, Abdulellah Abualshour, Ali Kassem Thabet, and Bernard Ghanem. Deepgcns: Making gcns go as deep as cnns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[4] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7:187–199, 2021.

[5] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.

[6] Chen Zhao, Jiaqi Yang, Xin Xiong, Angfan Zhu, Zhiguo Cao, and Xin Li. Rotation invariant point cloud classification: Where local geometry meets global topology. *arXiv preprint arXiv:1911.00195*, 2019.

[7] Shitong Luo, Jiahan Li, Jiaqi Guan, Yufeng Su, Chaoran Cheng, Jian Peng, and Jianzhu Ma. Equivariant point cloud analysis via learning orientations for message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18932–18941, 2022.

[8] Feiran Li, Kent Fujiwara, Fumio Okura, and Yasuyuki Matsushita. A closer look at rotation-invariant deep point cloud analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16218–16227, 2021.

[9] Kent Fujiwara and Taiichi Hashimoto. Neural implicit embedding for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11734–11743, 2020.

[10] 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*, pages 165–174, 2019.

[11] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18942–18952, 2022.

[12] Lulu Tang, Ke Chen, Chaozheng Wu, Yu Hong, Kui Jia, and Zhi-Xin Yang. Improving semantic analysis on point clouds via auxiliary supervision of local geometric priors. *IEEE Transactions on Cybernetics*, 52(6):4949–4959, 2020.

[13] Xiaomin Yue, Sophia Robert, and Leslie G Ungerleider. Curvature processing in human visual cortical areas. *NeuroImage*, 222:117295, 2020.

[14] Árni Kristjánsson and Peter Ulric Tse. Curvature discontinuities are cues for rapid shape analysis. *Perception & Psychophysics*, 63(3):390–403, 2001.

[15] Jeremy M Wolfe, Alice Yee, and Stacia R Friedman-Hill. Curvature is a basic feature for visual search tasks. *Perception*, 21(4):465–480, 1992.

[16] Robert L Fantz and Simon B Miranda. Newborn infant attention to form of contour. *Child Development*, pages 224–228, 1975.

[17] Rolandos Alexandros Potamias, Giorgos Bouritsas, and Stefanos Zafeiriou. Revisiting point cloud simplification: A learnable feature preserving approach. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 586–603. Springer, 2022.

[18] 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.

[19] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[20] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23192–23204, 2022.

[21] Can Qin, Haoxuan You, Lichen Wang, C.-C. Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019.

[22] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[23] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

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

[25] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, jun 1975.

[26] Jong-Chyi Su, Matheus Gadelha, Rui Wang, and Subhransu Maji. A deeper look at 3d shape classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[27] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3809–3820. PMLR, 18–24 Jul 2021.

[28] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.

[29] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[30] Shi Qiu, Saeed Anwar, and Nick Barnes. Pnp-3d: A plug-and-play for 3d point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1312–1319, 2023.

[31] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016.

[32] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 915–924, 2021.

[33] Ting Yao, Yehao Li, Yingwei Pan, and Tao Mei. Hgnet: Learning hierarchical geometry from points, edges, and surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21846–21855, 2023.

[34] Idan Achituve, Haggai Maron, and Gal Chechik. Self-supervised learning for domain adaptation on point clouds. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 123–133, 2021.

Acknowledgment

I extend my deepest gratitude to my advisor, Prof. Eunho Yang, for his insightful analyses and invaluable suggestions that have steered me towards conducting more robust research. My sincere appreciation also goes to my thesis committee members, Prof. Sung Ju Hwang and Prof. Jinwoo Shin, for their thoughtful comments on my research and the invaluable guidance on the essential attitude to uphold as a researcher.

A special thanks goes to my co-author and senior, Joonhyung. I have gained immense knowledge, from refining the topic to developing ideas and mastering the art of crafting a compelling paper. I am also grateful to all my lab members for their valuable advice, which has been instrumental in my growth throughout the Master's course.

Last but not least, I would like to thank my family – my father Yong-gun Kim, my mother Aehyun Kwon, and my younger sister Minseo Kim – who were my greatest supporters and always embraced me with love.

Curriculum Vitae

Name : Sohee Kim

Date of Birth : March 02, 1999

Educations

2022. 3. – 2024. 2. M.S. in Artificial Intelligence, KAIST

2018. 3. – 2022. 2. B.S. in School of Biomedical Engineering, Korea University
B.E. in Software Technology and Enterprise Program, Korea University

Publications

1. **S. Kim**, J. Park, and E. Yang, "PCS: Progressive Curvature Scanning for Contour-guided 3D Representation Learning." In Preparation.