This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Learnable Skeleton-Aware 3D Point Cloud Sampling

Cheng Wen* Baosheng Yu* Dacheng Tao
School of Computer Science, The University of Sydney, NSW 2008, Australia
cwen6671@uni.sydney.edu.au; baosheng.yu@sydney.edu.au; dacheng.tao@gmail.com

Abstract

Point cloud sampling is crucial for efficient large-scale point cloud analysis, where learning-to-sample methods have recently received increasing attention from the community for jointly training with downstream tasks. However, the above-mentioned task-specific sampling methods usually fail to explore the geometries of objects in an explicit manner. In this paper, we introduce a new skeleton-aware learning-to-sample method by learning object skeletons as the prior knowledge to preserve the object geometry and topology information during sampling. Specifically, without labor-intensive annotations per object category, we first learn category-agnostic object skeletons via the medial axis transform definition in an unsupervised manner. With object skeleton, we then evaluate the histogram of the local feature size as the prior knowledge to formulate skeleton-aware sampling from a probabilistic perspective. Additionally, the proposed skeleton-aware sampling pipeline with the task network is thus end-to-end trainable by exploring the reparameterization trick. Extensive experiments on three popular downstream tasks, point cloud classification, retrieval, and reconstruction, demonstrate the effectiveness of the proposed method for efficient point cloud analysis.

1. Introduction

With the development of 3D sensing technologies, acquiring 3D data becomes more accessible than before, and there are a growing number of data repositories available online, such as ModelNet [62], ShapeNet [6], ScanNet [11], and KITTI [18]. Among popular 3D shape representations such as point cloud [41], voxel [72], mesh [55], and multi-view images [51], the point cloud is becoming increasingly popular as the first-hand data captured by LiDAR or depth camera (e.g., Kinect), which has been widely applied in various applications such as scene reconstruction [19, 26], autonomous driving navigation [35], and virtual reality (VR) [59]. Though a high-resolution point cloud can accurately capture the geometry and topology details of complex objects, it remains challenging for those devices with limited computation and storage resources. Therefore, point cloud sampling, aiming to find a small set of points to represent the object shape and topology effectively, is usually indispensable for efficient large-scale point cloud analysis [24, 29, 32, 33, 41, 42, 57, 61, 71].

(a) (b) (c) (d)

Figure 1. (a) Point cloud; (b) Object skeleton; (c) Random sampling; (d) Skeleton-aware sampling. Compared with random sampling, skeleton-aware sampling tends to preserve the object geometry and topology information.

Traditional point cloud sampling methods such as random sampling (RS) and farthest point sampling (FPS) [15, 42] usually select a subset of points directly using the raw data information [7, 42, 43, 47, 68]. Specifically, RS is very efficient but may miss sparse regions, while FPS has better coverage on the entire point set but suffers from the latency bottleneck in parallel computation. To improve the performances on downstream tasks, learn-to-sample methods have been recently proposed to jointly optimize the sampling algorithm and each specific downstream task [10, 14, 20, 27, 56]. Though considerable progress has been achieved in downstream tasks such as point cloud classification and reconstruction, one critical issue remains poorly investigated: as objects usually have complex topology structures and irregular surface morphology, it is challenging to preserve the object's geometrical information during the point cloud sampling process.

Skeleton is an efficient representation to capture the underlying object shape structures, which has been widely used as the structural abstraction in many visual understanding tasks [31, 37, 53]. Inspired by this, we build our point cloud sampling strategy on top of the object skeleton to

*Equal contribution

preserve different objects' intrinsic geometrical and topological natures. Here we illustrate a comparison between random sampling and skeleton-aware sampling in Fig. 1. However, the skeleton extraction is usually non-trivial due to the following reasons [30, 52]: 1) given the diversity of object topological structures, it is difficult to have a consistent category-agnostic skeleton definition at the semantic level, where existing datasets usually consider only single or a few known object categories, such as human skeleton; 2) topological methods are usually category-agnostic by emphasizing geometrical and topological properties of the shape, such as its connectivity, topology, length, direction, and width. Nevertheless, they usually require a substantial amount of time for geometrical processing and are also notoriously sensitive to surface noise. Therefore, we resort to the medial axis transform (MAT) definition of object skeleton and deep neural networks to learn effective skeletal representations in an unsupervised manner.

With the learned object skeleton, we then formulate the skeleton-aware point cloud sampling pipeline from a probabilistic perspective. Specifically, we first calculate the local feature size (LFS) [2] for each point to measure the object's size near a particular point. We then explore the LFS distribution as the prior sampling probability on each point and use the LFS histogram in practice since per-point LFS is usually sensitive to point cloud noise. By learning object skeletons with deep neural networks, we have the skeleton-aware prior probability for sampling each point. To adapt skeleton-aware sampling for downstream tasks, we jointly optimize the posterior sampling probability on each point in an end-to-end manner. Notably, by exploring the categorical reparameterization with Gumbel-softmax [22], the categorical sampling based on LFS histogram is differentiable. Therefore, both sampling and task networks are end-to-end learnable for task-specific point cloud sampling. In this paper, our main contributions can be summarized as follows:

1. We propose a new skeleton-aware point cloud sampling method to preserve the object geometry and topology information during sampling.

2. We introduce the category-agnostic skeleton learning in an unsupervised manner to provide the prior knowledge for skeleton-aware point cloud sampling.

3. We conduct extensive experiments on three important downstream tasks, point cloud classification, retrieval, and reconstruction, to evaluate the proposed approach.

2. Related Work

Deep Learning on Point Clouds. Deep neural networks have been widely used in point cloud analysis, including point cloud classification/segmentation [29, 33, 41, 42, 57, 63], object detection/tracking [9, 48, 49, 72], point cloud autoencoders [1, 39, 65, 67], point cloud generation [1, 50, 58, 64], completion [5, 8, 21, 69] and registration [17, 38, 45]. A point cloud is usually not placed on a regular grid, and each point is independent of others. Meanwhile, the distances to neighboring points are not always fixed. Therefore, it is non-trivial to apply deep learning techniques to 3D point clouds. Specifically, PointNet [41] is the pioneering work in applying deep neural networks to point sets, which embeds the input into high dimensional space pointwisely and then uses a symmetric function to aggregate all point features in a permutation-invariant manner as the global features. The MLP block used in PointNet has become a fundamental component in many point cloud networks to learn pointwise representations. Recently, convolutional architectures have been further explored to aggregate local neighborhood hierarchical information and achieved superior performance on point cloud analysis [29, 54, 61].

Skeleton-Guided Learning. Object skeleton jointly describes an object's geometry, topology, and symmetry properties in a compact and intuitive way [52], which has benefited various tasks such as shape recognition [3], reconstruction [23, 53], segmentation [31] and point cloud completion [37, 70]. For example, [53] proposes a skeleton-aware deep learning approach to generate the mesh reconstruction of object surface a from single RGB image. [37] presents a learning method for point cloud completion via the shape skeleton. Existing methods directly use the ground truth skeletons as the supervision, which can be calculated by off-line algorithms like DPC [60], Q-MAT [28] or Coverage Axis [13]. However, as mentioned before, the skeleton computation is an ill-posed problem, meaning the computation is not steady and unique. Therefore, deep learning-based skeletal representations have been recently proposed [30, 44]. For example, [30] proposes to learn skeletal meshes from point clouds. Similarly, we also learn the skeleton in an unsupervised manner to facilitate our skeleton-aware sampling strategy.

Point Cloud Sampling. Point cloud sampling or simplification is essential, since it is always non-trivial to process high-resolution dense point clouds. Therefore, various methods have been explored to simplify dense point clouds [4, 7, 25, 34, 40, 43, 47]. For example, [47] first uses the K-means clustering to select representative points and the remove redundant points. [4] combines clustering and coarse-to-fine approaches for fast point cloud simplification. [34] presents an intrinsic point cloud simplification algorithm with density guarantee, which supports efficient uniform and feature-sensitive resampling. The above-mentioned point cloud simplification algorithms aim to find a subset of points in terms of geometry or topology, but fails to consider the downstream tasks during sampling. Several recent methods have proposed to learn to sample from

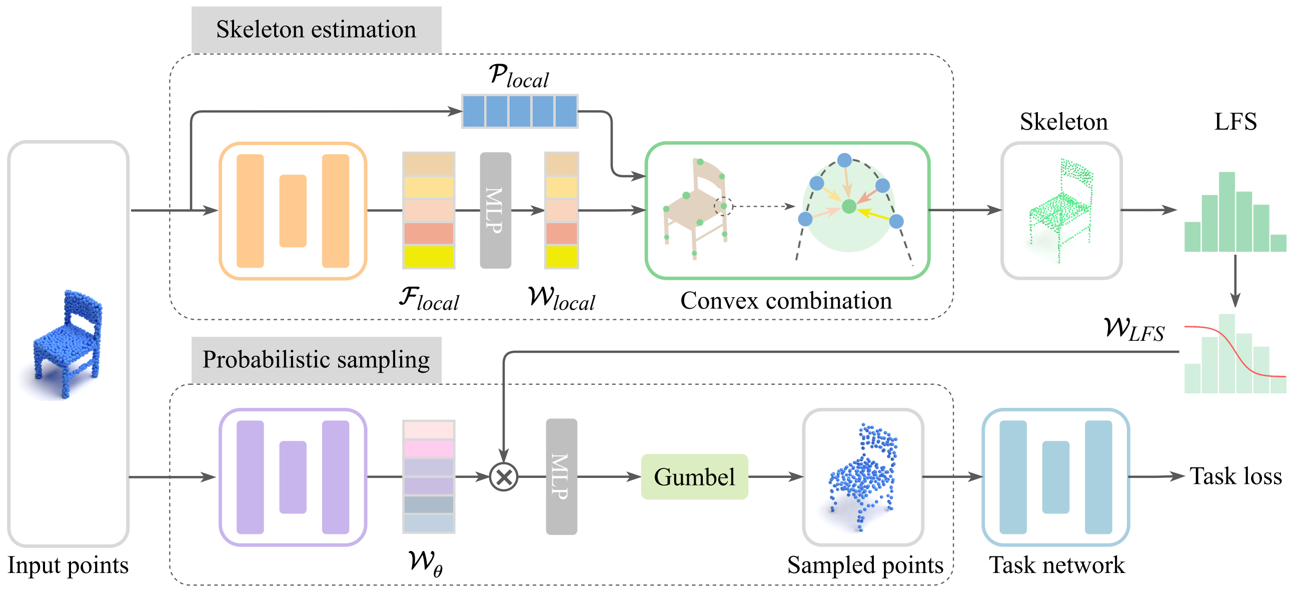


Figure 2. The main skeleton-aware point cloud sampling framework. Specifically, the whole pipeline consists of two main stages: 1) we first learn object skeleton in an unsupervised manner; 2) with the estimated skeleton, we evaluate the LFS distribution as the initial sampling weight to formulate the sampling process from a probabilistic perspective.

dense points [10,14,20,27,36,56,66]. Specifically, [14] and [27] have recently introduced learning-based task-oriented sampling strategies for the downstream tasks such as point cloud classification, retrieval, and reconstruction. [10] further proposes a learnable sampling and joint-training strategy to avoid overfitting on a specific task. [36] introduces a critical point layer to pass on points with the most functional features to the next layer and adaptively sample points. Differently, we not only consider the downstream tasks but also try to preserve the underlying geometrical and topological information in point clouds.

3. Method

In this section, we first provide an overview of skeleton-aware point cloud sampling. We then describe skeleton estimation and skeleton-aware probabilistic sampling.

3.1. Overview

The main skeleton-aware point cloud sampling framework is shown in Fig. 2. Given a point cloud $\mathcal{P}$, point cloud sampling aims to find a subset of points $\mathcal{P}_{sub} \subset \mathcal{P}$, where a good sampling strategy is expected that: 1) the selection is consistent regardless of any permutation of input points and outliers; 2) the selected points preserve the geometrical and topological information of the original point cloud; 3) the sampling process can be integrated and optimized jointly with downstream tasks to avoid significant performance degradation. Recently, learning-to-sample methods

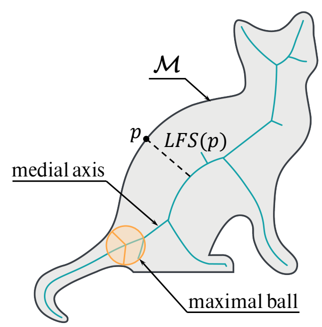
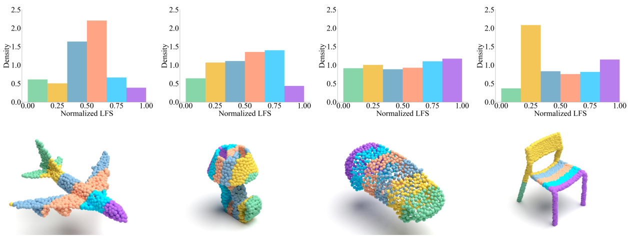


Figure 3. An illustration of the object with its medial axis in 2D space. Specifically, the medial axis is defined as the set of points with more than one closest point on the boundary surface, i.e., the centers of balls maximal inscribed in the object. LFS(p) indicates the Euclidean distance from the point p to the medial axis.

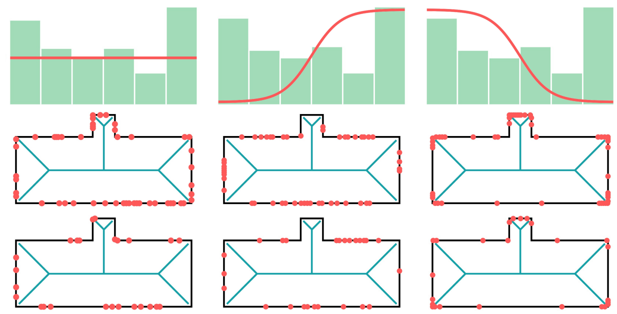
have achieved remarkable success by selecting task-specific points with end-to-end learning [10, 14, 20, 27, 36]. However, these methods neglect the complex object geometrical and topological structures during learning to sample. Therefore, we introduce object skeleton to explore the structure and topology information for point cloud sampling.

Object skeleton can be defined as a thin-centered structure that jointly describes the geometry and topology. Following the popular medial axis transform (MAT) definition [52], the object skeleton is characterized by a set of ball centers with the corresponding radii, where each ball is maximal inscribed in the object as shown in Fig. 3.



(a) Airplane (b) Lamp (c) Car (d) Chair

Figure 4. An illustration of the LFS distribution of different objects: airplane, lamp, car, and chair. Specifically, the object skeletons are computed using DPC [60] and the LFS values are normalized for a better comparison across different object categories.



(a) (b) (c)

Figure 5. The cyan line is the object's medial axis, and points (red) are sampled on the boundary. (a) Random sampling; (b) High sampling probability at large LFS area; and (c) High sampling probability at small LFS area. Top: LFS histogram and sampling weighting; Middle: 50 points; Bottom: 25 points.

Specifically, the skeleton is a compact shape representation with geometrical clues such as thickness, angles and branches aggregating. The structural and geometrical information significantly benefits complex 3D object analysis [31, 46]. To equip point cloud sampling with the skeleton-aware structural prior, we introduce the local feature size or LFS [2] to measure the size of the object near a particular point, e.g., the dashed line in Fig. 3. Notably, given a smooth manifold $\mathcal{M}$ and the point $p \in \mathcal{M}$, LFS(p) is defined as the Euclidean distance between p and the medial axis of $\mathcal{M}$ via the nearest point. Intuitively, LFS captures the surface property and how complicated $\mathcal{M}$ is locally distributed [12].

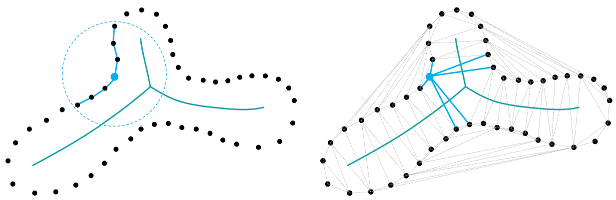
With the object skeleton, we then calculate the LFS histogram to form a robust skeleton-aware prior probability for sampling. Notably, LFS tends to distinguish the object parts with different geometrical structures, e.g., in Fig. 4, each LFS bin is closely related to a specific part of the object. Another important observation is that points with small LFS values are usually located at the delicate parts of the object, such as the areas of sharp edges or corners and these parts are usually informative and critical for geometrical analysis. For example, given the 2D object in Fig. 5, we sample points along the boundary with three different sampling strategies, random sampling, high sampling probability at large LFS area and high sampling probability at small LFS area. If we try to reconstruct the object using sampled points, the last strategy will have the smallest reconstruction error, i.e., points located on delicate parts are more important and the small LFS thus indicates importance. Consequently, we use this LFS histogram as the prior sampling probability. In the following subsection, we introduce the unsupervised skeleton estimation for efficient skeleton-aware point cloud sampling.

3.2. Unsupervised Skeleton Estimation

With the medial axis transform or MAT definition, we then have the category-agnostic object skeleton for point cloud sampling. However, it is non-trivial to efficiently apply the vanilla MAT algorithm due to its instability and algebraic complexity. Therefore, we introduce an unsupervised skeleton estimation method for point cloud sampling as follows. Given the input point cloud, it then aims to predict a set of K skeletal spheres, where each skeletal sphere is defined by the center (i.e., the skeleton point) $c_i \in \mathbb{R}^3$ and the radius $r_{c_i} \in \mathbb{R}$. That is, the skeletal spheres consist of two components, the coordinates of the sphere centers $\mathcal{C} \in \mathbb{R}^{K\times 3}$ and their radii $\mathcal{R} \in \mathbb{R}^K$. Generally, the 3D object skeleton contains both 1D curve segments and 2D surface sheets, which can represent underlying structures of various shapes. The rationale for representing the skeleton with sparse skeletal spheres lies in that the final goal of skeleton estimation is to calculate the LFS of each point. Additionally, as mentioned before, we will use the averaged LFS value in each bin instead of each individual LFS value. Therefore, dense LFS is not always necessary, which avoid introducing extra computation and storage cost for predicting dense skeletal points.

Following [30], we introduce our skeleton estimation algorithm as follows. Specifically, [30] uses PointNet++ [42] as the backbone, and the output of the last set abstraction layer is a set of points $\mathcal{P}_*$ and the corresponding contextual features $\mathcal{F}_*$. A skeletal point can always be considered the local center of a set of surface points. Consequently, the skeletal points can be generated by the convex combination of $\mathcal{P}_*$, i.e., $\mathcal{C} = \mathcal{W}_*^\top \mathcal{P}_*$, where $\mathcal{W}_*$ is the predicted weight matrix. The closest distances from $\mathcal{P}_*$ to the skeletal points $\mathcal{C}$ can be calculated and denoted as $\mathcal{D}_*$, and then the radii of all skeletal spheres are derived by $\mathcal{R} = \mathcal{W}_*^\top \mathcal{D}_*$. Similar to [30], we use the same loss function, which consists of three parts: a sampling loss $\mathcal{L}_s$, a point-to-sphere loss $\mathcal{L}_p$, and a radius regularizer $\mathcal{L}_r$, as follows:

$$\mathcal{L}_s = \sum_{p\in\mathcal{P}_*} \min_{t\in\mathcal{T}_*} \|p-t\|_2 + \sum_{t\in\mathcal{T}_*} \min_{p\in\mathcal{P}_*} \|t-p\|_2, \qquad (1)$$



(a) KNN neighbors (b) Delaunay neighbors

Figure 6. KNN neighbors and Delaunay neighbors. KNN always find the nearest neighbors while the Delaunay neighbors are these connected by Delaunay edges. Cyan line: object skeleton; Black points: surface points; Blue points: centroids. Local neighboring points are connected by blue line segments.

$$\mathcal{L}_p = \sum_{p\in\mathcal{P}_*}[\min_{c\in\mathcal{C}}\|p-c\|_2 - r_{c_p^{min}}] + \sum_{c\in\mathcal{C}}[\min_{p\in\mathcal{P}_*}\|c-p\|_2 - r_c], \tag{2}$$

$$\mathcal{L}_r = -\sum_{c\in\mathcal{C}} r_c, \tag{3}$$

where $\mathcal{T}_*$ is a set of points sampled on the surface of each skeletal sphere and c_p^{min} is the closest skeletal point to p.

Though the above-mentioned method can generate skeleton, each skeletal point is weighted by all points in $\mathcal{P}_*$. However, the skeleton prediction is always based on the local region, which means surface points far away from the skeletal point have no contribution to skeleton estimation. What's more, the convex combination requirement in [30] also poses a challenge to neural network learning since the network has to predict a large weight matrix. Therefore, we further propose a local weighting scheme concentrating on neighboring points to predict the skeleton. As shown in Fig. 2, our skeleton estimation network adopts the DGCNN [57] as the backbone to obtain the point-wise contextual features. Then we select K points as the centroids to group local points $\mathcal{P}_{local} \in \mathbb{R}^{K\times L\times 3}$ and features $\mathcal{F}_{local} \in \mathbb{R}^{K\times L\times D}$, where L is the number of neighbors of each centroid and D is the dimension of the features. Based on the grouped features, we will predict the weight matrix $\mathcal{W}_{local} \in \mathbb{R}^{K\times L\times 1}$ to derive the skeletal points, that is $\mathcal{C} = \mathcal{W}_{local}^{\top}\mathcal{P}_{local}$. Here, matrix transposition and multiplication are conducted on the last two dimensions.

A critical insight here is how to select neighboring points of each centroid. As presented in [12], the skeletal points can be estimated by Delaunay edge filtering. Inspired by this, we use the Delaunay neighbors instead of the KNN ones to predict the skeletal points. We illustrate the difference between KNN and Delaunay neighbors in Fig. 6. As shown, KNN always finds the nearest L neighbors in Euclidean space. However, if we aim to predict the skeletal points that are always centrally located in the object, it is beneficial to consider the antipodal surface points. That is the intuition to use the Delaunay neighbors. To select the Delaunay neighbors, we first apply the Delaunay triangulation and pick out these local points connected by Delaunay edges. With the estimated skeleton, we can calculate the LFS of each point. As LFS is sensitive to point cloud noise, the best practice in our work is to use the "smoothed" LFS, i.e., the LFS histogram. Given the LSF histogram, we can then have the midpoint of each bin $\bar{h}_1, \bar{h}_2, ..., \bar{h}_n$, where n is the number of bins. If the LFS of one point is in i-th bin, the initial sampling weight for the point is $e^{-\bar{h}_i}/\sum_i e^{-\bar{h}_i}$.

3.3. Skeleton-Aware Probabilistic Sampling

As mentioned before, we aim to find a subset of points that not only take care of the downstream tasks but also encode the geometrical structure information of 3D objects. To derive the sampling formulation, we treat the point sampling operation as drawing samples from the probability distribution with replacement. Specifically, given the input point cloud $\mathcal{P} = \{p_1, p_2, ..., p_N\}$, we already have the prior per-point sampling weight from the first stage, and this initial weight encapsulates the structural information that should be considered. After the point cloud goes through the neural network, we get the point-wise features. Therefore, the posterior sampling probability over point-wise features can be formulated as:

$$\mathscr{P}_\theta(p_1, ..., p_N) = \prod_{i=1}^{N}\sum_j \mathscr{P}_\theta(p_i|b_{ij})\mathscr{P}_{LFS}(b_{ij}), \tag{4}$$

where b_{ij} denotes the event that p_i locates in the j-th bin of the LFS histogram and the corresponding probability is $\mathscr{P}_{LFS}(b_{ij})$. The subscript θ indicates trainable parameters of the neural network. During training, we maximize the posterior over all training point clouds with respect to the individual point cloud feature and the network parameters.

Consequently, with the input point cloud $\mathcal{P}$, it is straightforward to consider points as different categories, and the task that samples a representative subset $\mathcal{P}_{sub} \subset \mathcal{P}$ then becomes to learn to draw samples from a discrete distribution, e.g. the categorical distribution. However, the sampling operation is usually non-differentiable. Though [14] and [27] are trainable, the sampled point is less interpretable from the probabilistic perspective, especially when sampling multiple points.

Categorical Reparameterization. Fortunately, Jang et al. [22] introduce an elegant reparameterization trick called Gumbel-softmax to enable discrete stochastic variables to back-propagate in neural network computation graphs. Therefore, we utilize the Gumbel-softmax trick to facilitate our sampling process. For a categorical distribution $Cat(c_1, ..., c_k)$ where k denotes the number of categories and c_i means the probability of category i, the Gumbel-softmax is designed as a discrete reparameterization trick to estimate smooth gradient with a continuous relaxation for the categorical variable. Given the Gumbel noise $\boldsymbol{g} = (g_1, ..., g_k)$ where $g_i \sim Gumbel(0, 1)$ is independent identically distributed, a soft categorical sample can be drawn by

$$\boldsymbol{y} = softmax((log(\boldsymbol{c}) + \boldsymbol{g})/\tau), \tag{5}$$

where $\boldsymbol{c} = (c_1, ..., c_k)$ and $\tau > 0$ is the annealing temperature. The equation above is referred as Gumbel-softmax, and as $\tau \to 0$, $\boldsymbol{y}$ will degenerate into the Gumbel-max form gradually,

$$\boldsymbol{y} = onehot(argmax((log(\boldsymbol{c}) + \boldsymbol{g}))), \tag{6}$$

which is an unbiased sample from $Cat(c_1, \ldots, c_k)$. In this way, we are able to draw differentiable samples from the distribution $Cat(c_1, \ldots, c_k)$ in the training phase. In practice, τ starts at a high value (e.g., 1.0) and anneals to a small value (e.g., 0.1). In the testing phase, discrete samples can be drawn with the Gumbel-max trick.

Therefore, we use a hard and discrete selection by exploring the trainable categorical reparameterization Gumbel-softmax as follows:

$$\mathcal{P}_{sub} = [Gs_\tau(Lp(\mathcal{W}_{LFS} \odot \mathcal{W}_\theta)]^\top \mathcal{P}, \tag{7}$$

where $\mathcal{P} \in \mathbb{R}^{N\times 3}$ is the input point cloud, $\mathcal{P}_{sub} \in \mathbb{R}^{M\times 3}$ is the sampled point cloud, and $Gs_\tau(\cdot)$ operation is the Gumbel-softmax trick with the annealing temperature τ. Additionally, $\mathcal{W}_{LFS} \in \mathbb{R}^{N\times 1}$ is calculated from the first stage and $\mathcal{W}_\theta \in \mathbb{R}^{N\times D_{sp}}$ is feature matrix from the sampling network, where D_{sp} indicates the dimension of the features. $Lp(\cdot) : \mathbb{R}^{N\times D_{sp}} \to \mathbb{R}^{N\times M}$ is the linear mapping layer. $\odot$ indicates the element-wise Hadamard product operation. In practice, we start with a large τ such as $\tau = 1.0$ and gradually decrease it. In the training phase, it provides smooth gradients using the discrete reparameterization trick. With annealing, it degenerates to a hard selection in the testing phase.

4. Experiments

In this section, we evaluate the proposed skeleton-aware sampling method by applying it to three downstream tasks: point cloud classification, retrieval, and reconstruction. All experiments are conducted on the widely used ModelNet40 [62] dataset, and we use the official split of 9,843 training and 2,468 testing samples.

4.1. Implementation Details

We implement the proposed method using PyTorch, and we will give the detailed network architecture in our supplementary materials. Our network training consists of two stages. We train the skeleton estimation network in the first stage to get the point cloud skeleton. The number of predicted skeletal spheres is 128, and the number of bins for the LFS histogram is 6. In the second stage, we train the sampling network using task losses. In both stages, we use the Adam optimizer with a learning rate of 0.001.

Sampling Ratio	Classification Acc(%)				
	FPS [42]	S-NET [14]	SN [27]	MS [10]	Ours
8	70.4	77.5	83.7	88.0	**89.1**
16	46.3	70.4	82.2	85.5	**88.8**
32	26.3	60.6	80.1	81.5	**87.4**
64	13.5	36.1	54.1	61.6	**82.9**

Table 1. Point cloud classification results on ModelNet40. The task network is trained on complete data with 1024 points.

4.2. Point Cloud Classification

To evaluate the proposed method for point cloud classification, we employ the full version of PointNet [41] as the task network. The classification accuracy of PointNet is trained on the complete data (1024 points per point cloud without normals) and tested on sampled points of different sizes. The evaluation metric is overall accuracy (OA), and for a fair comparison, the upper bound accuracy trained on 1024 points is 89.5% (the same as MetaSampler [10]). We compare with four methods, FPS [42], S-NET [14], SampleNet [27] and MetaSampler [10]. We report the classification results of four different sampling ratios in Table 1, where the proposed method outperforms recent state-of-the-art methods. Specifically, the classification accuracy achieved by the proposed approach at the sampling ratio of 8 is very close to the upper bound accuracy (89.1% *vs.* 89.5%), which shows the effectiveness of our skeleton-aware strategy. We also notice that the performance gap between the proposed method and others is becoming large when increasing the sampling rate (e.g., at the sampling ratio of 64 with 16 points left). This further demonstrates the superiority of the proposed method in preserving task-significant features.

Ablation Study on Skeleton-Aware Prior. We show how the prior sampling weight operates by comparing with two other commonly used non-learned sampling methods, random sampling (RS) and farthest point sampling (FPS). In Fig. 7, for visualization purposes, we use 10,000 points as input, and the number of sampled points is 1,024. Generally, RS randomly picks 1024 points from the original data, and FPS selects a subset of points far from each other. Since both two methods do not consider any geometrical information, these two sampling methods do not change the LFS distribution. Differently, we propose the skeleton-aware sampling in this work, which is able to use the LFS histogram as the initial sampling weight and also learn to sample for downstream tasks. That is, we will pay more attention to those points with small LFS values. As shown

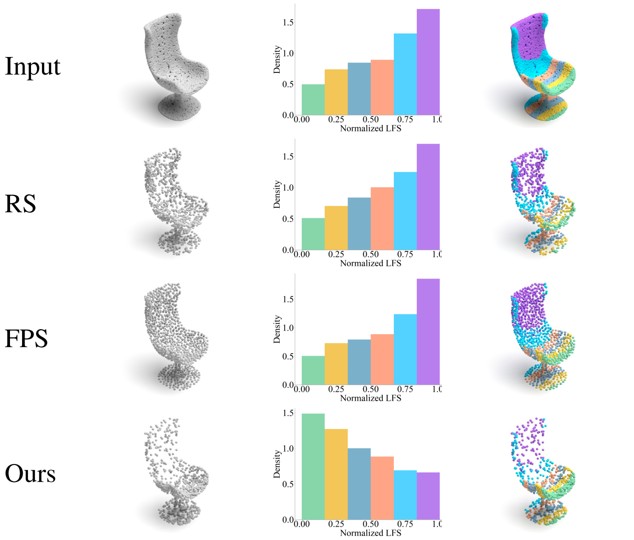


Figure 7. Visualization of different sampling strategies. RS and FPS do not encode geometrical information to change the LFS distribution after sampling, while the proposed considers the LFS histogram to obtain informative skeleton-aware sampling.

in Fig. 7, the RS and FPS keep nearly the same LFS distribution with the original point cloud after sampling, but our method encodes the geometrical information and gives out the different LFS distribution. The final results can also prove this by our sampled points concentrated on small LFS areas. In addition, we report the classification results without the LFS weight in Table 2. Here, "GT" means we use the ground truth skeletons computed by DPC [60] instead of the estimated ones from the first stage. As shown, using the skeleton-aware strategy as the initial sampling weight can always benefit the following learning task. Moreover, our method achieves comparable results with that using ground truth skeletons.

Sampling Ratio	Classification Acc(%)		
	GT	Ours(w/o)	Ours
8	89.1	88.2	89.1
16	88.8	87.4	88.8
32	87.5	85.9	87.4
64	82.8	80.3	82.9

Table 2. Point cloud classification results without the initial LFS weight. GT indicates using the ground truth skeletons generated by DPC [60].

Ablation Study on LFS Histogram. The number of bins reveals the LFS distribution at different levels. Using wider bins reduces the LFS noise, while using narrower bins gives better precision to the LFS density estimation. Thus, varying the bin number within the LFS histogram can be beneficial. We report the classification results with different bins in Table 3. The aggressive setting is to use only one bin, and this case is equivalent to no initial LSF weight applied. Generally, we find that six bins are suitable for most cases.

Sampling Ratio	Bins				
	1	3	6	12	18
8	88.2	88.4	89.1	89.1	89.0
16	87.4	88.0	88.8	88.5	88.8
32	85.9	86.6	87.4	87.2	87.4
64	80.3	81.2	82.9	82.8	82.7

Table 3. Point cloud classification results with different bins. Generally, we find that six bins already work well in most cases.

Ablation Study on Skeletal Spheres. As mentioned before, the object's skeleton is represented with a set of skeletal spheres. Generally, increasing the number of predicted skeletal spheres will lead to detailed structures. Thus, it is beneficial to disclose how this hyper-parameter influences the following tasks. We show the classification results in Table 4 with different numbers of skeletal spheres predicted. As shown, our method is not sensitive to the parameter and achieves steady performance.

Sampling Ratio	Skeletal Spheres			
	32	64	128	256
8	87.2	88.4	89.1	89.1
16	85.8	87.1	88.8	88.5
32	84.0	85.1	87.4	87.2
64	79.3	80.5	82.9	82.8

Table 4. Point cloud classification results when using different numbers of skeletal spheres.

Ablation Study on Annealing Strategy. As we apply the Gumbel-softmax operation in our method, we gradually anneal the temperature τ to facilitate our network training in the second stage. Thus, it is beneficial to evaluate the classification performance when adopting different annealing strategies. During training, we let $\tau = 1.0$ to allow gradients past the samples and then gradually anneal the temperature until $\tau = 0.5$ (but not completely to 0, as the gradients would blow up). In the test phase, as $\tau \to 0.0$, the softmax becomes an argmax and the Gumbel-softmax distribution becomes the categorical distribution. We report the classification results in Table. 5 with four different annealing strategies, including constant, step, linear and exponential. Here, the "constant" means τ keeps being 1.0 without annealing during training.

Temperature (τ)	Sampling Ratio			
	8	16	32	64
Constant	87.7	86.4	85.1	81.1
Step	88.8	87.2	86.6	82.1
Linear rectified	89.0	87.9	87.6	82.3
Exponential	89.1	88.8	87.4	82.9

Table 5. Point cloud classification results for using different annealing strategies. Here, the "Constant" row means we do not anneal the τ and keep $\tau = 1.0$ during training.

Model Complexity. In Table 6, we report the model complexity of the proposed method. Specifically, the FLOPs increase with a large number of predicted skeletal spheres. The number of parameters for the probabilistic sampling network is 1.24M, and the number of FLOPs is 2.93G at the sampling ratio of 8.

Model Complexity		Params (M)	FLOPs (G)
Skeletal Spheres	32	1.54	3.43
	64	1.54	3.86
	128	1.54	4.71
	256	1.54	6.41
Sampling Ratio	8	1.24	2.93
	16	1.23	2.90
	32	1.22	2.88
	64	1.21	2.86

Table 6. The model complexity of the proposed skeleton estimation network.

4.3. Point Cloud Retrieval

In this subsection, we perform point cloud retrieval experiments to further demonstrate the effectiveness of the proposed method for downstream tasks. We follow the same evaluation setting in [10, 14, 27] to report the performance on the point cloud retrieval. Specifically, we take our method trained with PointNet [41] for classification as the task network without re-training it. We sample points of different sizes fed into PointNet and use its penultimate layer as the shape descriptor. Retrieval is done based on the L2 distance on the shape descriptor, and the evaluation metric is the mean Average Precision (mAP). In Table. 7, we summarize the performances of different methods and observe that our method consistently outperforms others by a large margin at different sampling ratios.

4.4. Point Cloud Reconstruction

In this subsection, we perform point cloud reconstruction experiments to further demonstrate the effectiveness of the proposed method for downstream tasks. We use an autoencoder as the task network on the ModelNet40 [62] dataset. Following [10], we employ the Point Completion Network (PCN) [69] to minimize the Chamfer distance (CD) [16] between the input and output points. As demonstrated in Table 8, our method outperforms others and the improvement is consistent across all sampling ratios, further exhibiting the effectiveness of our method.

Sampling Ratio	Retrieval Performance (mAP)			
	FPS [42]	S-NET [14]	SN [27]	Ours
8	58.3	60.4	68.8	**72.2**
16	49.4	59.0	65.2	**70.9**
32	37.7	59.0	62.5	**67.1**
64	27.4	54.5	59.5	**62.6**

Table 7. Point cloud retrieval results on the ModelNet40 dataset. Specifically, the proposed method achieves superior performance across different sampling ratios.

Sampling Ratio	Reconstruction Performance (CD)			
	FPS [42]	SN [27]	MS [10]	Ours
8	3.78	3.29	3.05	**2.97**
16	4.03	3.32	3.15	**3.01**
32	4.25	3.61	3.37	**3.24**
64	4.78	4.43	4.31	**3.78**

Table 8. Point cloud reconstruction results on the ModelNet40 dataset. Specifically, the original size of data is 1024 points.

5. Conclusion

In this paper, we propose a skeleton-aware learning-to-sample method for point cloud sampling. The skeletal representation of the point cloud is learned in an unsupervised manner to avoid label-intensive annotations. Given the learned skeleton, the skeleton-aware point cloud sampling process is conducted in a probabilistic way according to the LFS distribution. The proposed skeleton-aware sampling method is end-to-end trainable. To evaluate the proposed method, we perform experiments on several downstream point cloud analysis tasks such as classification, retrieval, and reconstruction, and the proposed method consistently outperforms traditional and recent learning-to-sample methods by a large margin.

Acknowledgement

Mr Cheng Wen and Dr Baosheng Yu were supported by Australian Research Council Projects in part by IH180100002 and FL170100117.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International Conference on Machine Learning (ICML)*, pages 40–49. PMLR, 2018. 2

[2] Nina Amenta and Marshall Bern. Surface reconstruction by voronoi filtering. In *Proceedings of the 14th Annual Symposium on Computational Geometry*, pages 39–48, 1998. 2, 4

[3] Xiang Bai, Xinggang Wang, Longin Jan Latecki, Wenyu Liu, and Zhuowen Tu. Active skeleton for non-rigid object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 575–582, 2009. 2

[4] Halim Benhabiles, Olivier Aubreton, Hichem Barki, and Hedi Tabia. Fast simplification with sharp feature preserving for 3d point clouds. In *2013 11th International Symposium on Programming and Systems (ISPS)*, pages 47–52. IEEE, 2013. 2

[5] Yingjie Cai, Kwan-Yee Lin, Chao Zhang, Qiang Wang, Xiaogang Wang, and Hongsheng Li. Learning a structured latent space for unsupervised point cloud completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5543–5553, 2022. 2

[6] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, et al. Shapenet: An information-rich 3d model repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University, Princeton University, Toyota Technological Institute at Chicago, 2015. 1

[7] Siheng Chen, Dong Tian, Chen Feng, Anthony Vetro, and Jelena Kovačević. Fast resampling of three-dimensional point clouds via graphs. *IEEE Transactions on Signal Processing*, 66(3):666–681, 2017. 1, 2

[8] Xuelin Chen, Baoquan Chen, and Niloy J Mitra. Unpaired point cloud completion on real scans using adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020. 2

[9] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1907–1915, 2017. 2

[10] Ta-Ying Cheng, Qingyong Hu, Qian Xie, Niki Trigoni, and Andrew Markham. Meta-sampler: Almost-universal yet task-oriented sampling for point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 694–710, 2022. 1, 3, 6, 8

[11] 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 (CVPR)*, pages 5828–5839, 2017. 1

[12] Tamal K Dey and Wulue Zhao. Approximating the medial axis from the voronoi diagram with a convergence guarantee. *Algorithmica*, 38(1):179–200, 2004. 4, 5

[13] Zhiyang Dou, Cheng Lin, Rui Xu, Lei Yang, Shiqing Xin, Taku Komura, and Wenping Wang. Coverage axis: Inner point selection for 3d shape skeletonization. In *Computer Graphics Forum*, volume 41, pages 419–432. Wiley Online Library, 2022. 2

[14] Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sample. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2760–2769, 2019. 1, 3, 5, 6, 8

[15] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing (TIP)*, 6(9):1305–1315, 1997. 1

[16] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 605–613, 2017. 8

[17] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8893–8902, 2021. 2

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 1

[19] Ruizhen Hu, Cheng Wen, Oliver Van Kaick, Luanmin Chen, Di Lin, Daniel Cohen-Or, and Hui Huang. Semantic object reconstruction via casual handheld scanning. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018. 1

[20] Tianxin Huang, Jiangning Zhang, Jun Chen, Yuang Liu, and Yong Liu. Resolution-free point cloud sampling network with data distillation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 3

[21] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7662–7670, 2020. 2

[22] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017. 2, 5

[23] Haiyong Jiang, Jianfei Cai, and Jianmin Zheng. Skeleton-aware 3d human shape reconstruction from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5431–5441, 2019. 2

[24] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Amalgamating knowledge from heterogeneous graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15709–15718, 2021. 1

[25] Sagi Katz and Ayellet Tal. Improving the visual comprehension of point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 121–128, 2013. 2

[26] Ziquan Lan, Zi Jian Yew, and Gim Hee Lee. Robust point cloud based reconstruction of large-scale outdoor scenes. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 9690–9698, 2019. 1

[27] Itai Lang, Asaf Manor, and Shai Avidan. Samplenet: Differentiable point cloud sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7578–7588, 2020. 1, 3, 5, 6, 8

[28] Pan Li, Bin Wang, Feng Sun, Xiaohu Guo, Caiming Zhang, and Wenping Wang. Q-mat: Computing medial axis transform by quadratic error minimization. *ACM Transactions on Graphics (TOG)*, 35(1):1–16, 2015. 2

[29] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 820–830, 2018. 1, 2

[30] Cheng Lin, Changjian Li, Yuan Liu, Nenglun Chen, Yi-King Choi, and Wenping Wang. Point2skeleton: Learning skeletal representations from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4277–4286, 2021. 2, 4, 5

[31] Cheng Lin, Lingjie Liu, Changjian Li, Leif Kobbelt, Bin Wang, Shiqing Xin, and Wenping Wang. Seg-mat: 3d shape segmentation using medial axis transform. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2020. 1, 2, 4

[32] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8895–8904, 2019. 1

[33] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2

[34] Carsten Moenning and Neil A Dodgson. Intrinsic point cloud simplification. *Proc. 14th GrahiCon*, 14:23, 2004. 2

[35] Balázs Nagy and Csaba Benedek. Real-time point cloud alignment for vehicle localization in a high resolution 3d map. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 1

[36] Ehsan Nezhadarya, Ehsan Taghavi, Ryan Razani, Bingbing Liu, and Jun Luo. Adaptive hierarchical down-sampling for point cloud classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12956–12964, 2020. 3

[37] Yinyu Nie, Yiqun Lin, Xiaoguang Han, Shihui Guo, Jian Chang, Shuguang Cui, Jian Zhang, et al. Skeleton-bridged point completion: From global inference to local adjustment. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 16119–16130, 2020. 1, 2

[38] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 3dregnet: A deep neural network for 3d point registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7193–7203, 2020. 2

[39] Jiahao Pang, Duanshun Li, and Dong Tian. Tearingnet: Point cloud autoencoder to learn topology-friendly representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7453–7462, 2021. 2

[40] Mark Pauly, Markus Gross, and Leif P Kobbelt. Efficient simplification of point-sampled surfaces. In *IEEE Visualization, 2002. VIS 2002.*, pages 163–170. IEEE, 2002. 2

[41] 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)*, pages 652–660, 2017. 1, 2, 6, 8

[42] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5099–5108, 2017. 1, 2, 4, 6, 8

[43] Junkun Qi, Wei Hu, and Zongming Guo. Feature preserving and uniformity-controllable point cloud simplification on graph. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 284–289. IEEE, 2019. 1, 2

[44] Daniel Rebain, Ke Li, Vincent Sitzmann, Soroosh Yazdani, Kwang Moo Yi, and Andrea Tagliasacchi. Deep medial fields. *CoRR*, 2021. 2

[45] Vinit Sarode, Xueqian Li, Hunter Goforth, Yasuhiro Aoki, Rangaprasad Arun Srivatsan, Simon Lucey, and Howie Choset. Pcrnet: point cloud registration network using pointnet encoding. *CoRR*, 2019. 2

[46] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249–259, 2008. 4

[47] Bao-Quan Shi, Jin Liang, and Qing Liu. Adaptive simplification of point cloud using k-means clustering. *Computer-Aided Design*, 43(8):910–922, 2011. 1, 2

[48] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019. 2

[49] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1711–1719, 2020. 2

[50] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3859–3868, 2019. 2

[51] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 945–953, 2015. 1

[52] Andrea Tagliasacchi, Thomas Delame, Michela Spagnuolo, Nina Amenta, and Alexandru Telea. 3d skeletons: A state-of-the-art report. In *Computer Graphics Forum*, volume 35, pages 573–597. Wiley Online Library, 2016. 2, 3

[53] Jiapeng Tang, Xiaoguang Han, Junyi Pan, Kui Jia, and Xin Tong. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4541–4550, 2019. 1, 2

[54] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6411–6420, 2019. 2

[55] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 1

[56] Xu Wang, Yi Jin, Yigang Cen, Tao Wang, Bowen Tang, and Yidong Li. Lightn: Light-weight transformer network for performance-overhead tradeoff in point cloud downsampling. *CoRR*, 2022. 1, 3

[57] 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. 1, 2, 5

[58] Cheng Wen, Baosheng Yu, and Dacheng Tao. Learning progressive point embeddings for 3d point cloud generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10266–10275, 2021. 2

[59] Florian Wirth, Jannik Quchl, Jeffrey Ota, and Christoph Stiller. Pointatme: Efficient 3d point cloud labeling in virtual reality. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1693–1698. IEEE, 2019. 1

[60] Shihao Wu, Hui Huang, Minglun Gong, Matthias Zwicker, and Daniel Cohen-Or. Deep points consolidation. *ACM Transactions on Graphics (TOG)*, 34(6):1–13, 2015. 2, 4, 7

[61] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9621–9630, 2019. 1, 2

[62] 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 (CVPR)*, pages 1912–1920, 2015. 1, 6, 8

[63] 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 International Conference on Computer Vision (ICCV)*, pages 915–924, 2021. 2

[64] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4541–4550, 2019. 2

[65] Juyoung Yang, Pyunghwan Ahn, Doyeon Kim, Haeil Lee, and Junmo Kim. Progressive seed generation auto-encoder for unsupervised point cloud learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6413–6422, 2021. 2

[66] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3323–3332, 2019. 3

[67] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 206–215, 2018. 2

[68] Zhiwen Yu, Hau-San Wong, Hong Peng, and Qianli Ma. Asm: An adaptive simplification method for 3d point-based models. *Computer-Aided Design*, 42(7):598–612, 2010. 1

[69] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018. 2, 8

[70] Wenxiao Zhang, Zhen Dong, Jun Liu, Qingan Yan, Chunxia Xiao, et al. Point cloud completion via skeleton-detail transformer. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2022. 2

[71] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 16259–16268, 2021. 1

[72] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499, 2018. 1, 2