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].

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
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, 55], 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
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 $\mathcal{P}_{\text{sub}} \subset \mathcal{P}$, 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 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.

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.

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.
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 $M$ and the point $p \in M$, LFS$(p)$ is defined as the Euclidean distance between $p$ and the medial axis of $M$ via the nearest point. Intuitively, LFS captures the surface property and how complicated $M$ 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 $C \in \mathbb{R}^{K \times 3}$ and their radii $R \in \mathbb{R}^K$. 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 $P_s$ and the corresponding contextual features $F_s$. 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 $P_s$, i.e., $C = W^s X^s$, where $W_s$ is the predicted weight matrix. The closest distances from $P_s$ to the skeletal points $C$ can be calculated and denoted as $D_s$, and then the radii of all skeletal spheres are derived by $R = W^s D_s$. Similar to [30], we use the same loss function, which consists of three parts: a sampling loss $L_s$, a point-to-sphere loss $L_p$, and a radius regularizer $L_r$, as follows:

$$L_s = \sum_{p \in P_s} \min_{t \in T_s} \|p - t\|_2 + \sum_{t \in T_s} \min_{p \in P_s} \|t - p\|_2,$$  \hspace{1cm} (1)
Figure 6. KNN neighbors and Delaunay neighbors. KNN always find the nearest points 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.

\[ L_p = \sum_{p \in P_*} \left[ \min_{c \in C} \| p - c \|_2 - r_{c_{\min}} \right] \]
\[ + \sum_{c \in C \cap p \in P_*} \left[ \min_{i, j} \| c - p \|_2 - r_{c_i} \right] \]
\[ L_r = -\sum_{c \in C} r_{c_i} \]

where \( P_* \) is a set of points sampled on the surface of each skeletal sphere and \( r_{c_{\min}} \) is the closest skeletal point to \( p \).

Though the above-mentioned method can generate skeleton, each skeletal point is weighted by all points in \( P_* \). 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, the posterior sampling probability over point-wise features can be formulated as:

\[ P(\theta(p_1, ..., p_N)) = \prod_{i=1}^{N} \sum_{j} P(\theta(p_i | b_{ij}) \cdot P_{LFS}(b_{ij}), \]

where \( b_{ij} \) denotes the event that \( p_i \) locates in the \( j \)-th bin of the LFS histogram and the corresponding probability is \( P_{LFS}(b_{ij}) \). The subcript \( \theta \) 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 \( g = (g_1, ..., g_k) \) where \( g_i \sim \text{Gumbel}(0, 1) \) is independent identically distributed, a soft categorical sample can be drawn by
\[
y = \text{softmax}((\log(c) + g)/\tau), \tag{5}
\]
where \( c = (c_1, ..., c_k) \) and \( \tau > 0 \) is the annealing temperature. The equation above is referred as Gumbel-softmax, and as \( \tau \to 0 \), \( y \) will degenerate into the Gumbel-max form gradually,
\[
y = \text{onehot}(\argmax((\log(c) + g))), \tag{6}
\]
which is an unbiased sample from \( \text{Cat}(c_1, ..., c_k) \). In this way, we are able to draw differentiable samples from the distribution \( \text{Cat}(c_1, ..., c_k) \) in the training phase. In practice, \( \tau \) 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:
\[
\mathcal{P}_{\text{sub}} = [G_{s_e}(Lp(W_{LFS} \odot W_\theta))]^\top \mathcal{P}, \tag{7}
\]
where \( \mathcal{P} \in \mathbb{R}^{N \times 3} \) is the input point cloud, \( \mathcal{P}_{\text{sub}} \in \mathbb{R}^{M \times 3} \) is the sampled point cloud, and \( G_{s_e}(\cdot) \) is the Gumbel-softmax trick with the annealing temperature \( \tau \). Additionally, \( W_{LFS} \in \mathbb{R}^{N \times 1} \) is calculated from the first stage and \( W_\theta \in \mathbb{R}^{N \times D_{sp}} \) 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 \( \tau \) 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 degrades to a hard selection in the testing phase.

\begin{table}[ht]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
\hline
8 & 70.4 & 77.5 & 83.7 & 88.0 & \textbf{89.1} \\
16 & 46.3 & 70.4 & 82.2 & 85.5 & \textbf{88.8} \\
32 & 26.3 & 60.6 & 80.1 & 81.5 & \textbf{87.4} \\
64 & 13.5 & 36.1 & 54.1 & 61.6 & \textbf{82.9} \\
\hline
\end{tabular}
\caption{Point cloud classification results on ModelNet40. The task network is trained on complete data with 1024 points.}
\end{table}

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.

\section*{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
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, 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.

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.

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.

Ablation Study on Annealing Strategy. As we apply the Gumbel-softmax operation in our method, we gradually anneal the temperature $\tau$ 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 $\tau$ keeps being 1.0 without annealing during training.
### Table 5. Point cloud classification results for using different annealing strategies. Here, the “Constant” row means we do not anneal the $\tau$ and keep $\tau = 1.0$ during training.

<table>
<thead>
<tr>
<th>Temperature ($\tau$)</th>
<th>Sampling Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>8 16 32 64</td>
</tr>
<tr>
<td>Step</td>
<td>87.7 86.4 85.1 81.1</td>
</tr>
<tr>
<td>Linear rectified</td>
<td>89.0 87.9 87.6 82.3</td>
</tr>
<tr>
<td>Exponential</td>
<td>89.1 88.8 87.4 82.9</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Model Complexity</th>
<th>Params (M)</th>
<th>FLOPs (G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skeletal</td>
<td>32</td>
<td>1.54</td>
</tr>
<tr>
<td>Spheres</td>
<td>64</td>
<td>1.54</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>1.54</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>1.54</td>
</tr>
<tr>
<td>Sampling Ratio</td>
<td>8</td>
<td>1.24</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>1.23</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.22</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>1.21</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Sampling Ratio</th>
<th>Retrieval Performance (mAP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>58.3</td>
</tr>
<tr>
<td>16</td>
<td>49.4</td>
</tr>
<tr>
<td>32</td>
<td>37.7</td>
</tr>
<tr>
<td>64</td>
<td>27.4</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Sampling Ratio</th>
<th>Reconstruction Performance (CD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>3.78</td>
</tr>
<tr>
<td>16</td>
<td>4.03</td>
</tr>
<tr>
<td>32</td>
<td>4.25</td>
</tr>
<tr>
<td>64</td>
<td>4.78</td>
</tr>
</tbody>
</table>

### 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 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.

### 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.

### 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


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


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


