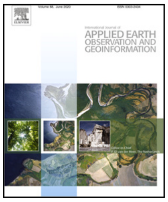
Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

journal homepage: www.elsevier.com/locate/jag



Dynamic clustering transformer network for point cloud segmentation

Dening Lu [a], Jun Zhou [b], Kyle (Yilin) Gao [a], Jing Du [a], Linlin Xu [a,*], Jonathan Li [c]

[a] *The Department of Systems Design Engineering, University of Waterloo, Waterloo, N2L3G1, Canada*
[b] *School of Nursing, The Hong Kong Polytechnic University, Hong Kong, TU428, China*
[c] *The Department of Geography and Environmental Management, University of Waterloo, Waterloo, N2L3G1, Canada*

ARTICLE INFO

Keywords:
3D transformer
Hierarchical data processing
Point cloud segmentation
Deep learning
Dynamic clustering

ABSTRACT

Point cloud segmentation is one of the most important tasks in LiDAR remote sensing with widespread scientific, industrial, and commercial applications. The research thereof has resulted in many breakthroughs in 3D object and scene understanding. Existing methods typically utilize hierarchical architectures for feature representation. However, the commonly used sampling and grouping methods in hierarchical networks are not only time-consuming but also limited to point-wise 3D coordinates, ignoring the local semantic homogeneity of point clusters. To address these issues, we propose a novel 3D point cloud representation network, called Dynamic Clustering Transformer Network (DCTNet). It has an encoder–decoder architecture, allowing for both local and global feature learning. Specifically, the encoder consists of a series of dynamic clustering-based Local Feature Aggregating (LFA) blocks and Transformer-based Global Feature Learning (GFL) blocks. In the LFA block, we propose novel semantic feature-based dynamic sampling and clustering methods, which enable the model to be aware of local semantic homogeneity for local feature aggregation. Furthermore, instead of traditional interpolation approaches, we propose a new semantic feature-guided upsampling method in the decoder for dense prediction. To our knowledge, DCTNet is the first work to introduce semantic information-based dynamic clustering into 3D Transformers. Extensive experiments on an object-based dataset (ShapeNet), and an airborne multispectral LiDAR dataset demonstrate the State-of-the-Art (SOTA) segmentation performance of DCTNet in terms of both accuracy and efficiency. *Our code will be made publicly available.*

1. Introduction

Semantic segmentation of 3D point clouds in LiDAR remote sensing is pivotal for creating highly detailed models of the Earth's surface, facilitating precise terrain analysis and vegetation characterization (Qin et al., 2023; Chen et al., 2023; Wei et al., 2023; Bui and Glennie, 2023; Tao et al., 2022). It plays a crucial role in diverse applications, ranging from urban planning, disaster response, and environmental monitoring to infrastructure management, offering invaluable insights through enhanced spatial visualization and analysis (Xiao et al., 2023; Chen and Cho, 2022; Zováthi et al., 2022; Li et al., 2022; Lin and Habib, 2022). Existing methods for 3D point cloud segmentation can be generally divided into three categories: view-based (Kundu et al., 2020; Robert et al., 2022; Mascaro et al., 2021; Antonello et al., 2018; Dai and Nießner, 2018), voxel-based (Maturana and Scherer, Sep. 2015; Riegler et al., 2017; Zhou and Tuzel, 2018; Zhang et al., 2022b), and point-based (Charles et al., 2017; Qi et al., 2017; Wang et al., 2019b; Thomas et al., 2019; Zhao et al., 2021b; Guo et al., 2021; Lai et al., 2022). Most of them utilized hierarchical structures for point cloud processing, focusing on local feature extraction but often ignoring long-range context dependency modeling (Lai et al., 2022).

The hierarchical structure typically involves two key steps: point cloud sampling and grouping. Currently, most hierarchical point cloud processing methods use the Farthest Point Sampling (FPS) (Qi et al., 2017) algorithm, sampling points evenly across the geometric space. However, FPS only focuses on the geometric properties of point clouds, ignoring their semantic features. This causes neural networks to de-emphasize some fine-level object parts with significant semantic information. Moreover, FPS is very time-consuming, often causing a computational bottleneck. Additionally, after downsampling, k-Nearest Neighborhood (kNN) and ball query (Qi et al., 2017) are widely used for the point cloud grouping. However, such grouping methods are still strictly based on the geometric properties of points. In this situation, the local feature aggregation tends to be disturbed by semantic heterogeneity in local neighborhoods, especially for points at the boundaries of adjacent parts. The aforementioned deficiencies are particularly pronounced when dealing with large-scale LiDAR datasets comprising

* Corresponding author.
E-mail addresses: d62lu@uwaterloo.ca (D. Lu), zachary-jun.zhou@connect.polyu.hk (J. Zhou), y56gao@uwaterloo.ca (K.(Y. Gao), j7du@uwaterloo.ca (J. Du), l44xu@uwaterloo.ca (L. Xu), junli@uwaterloo.ca (J. Li).

https://doi.org/10.1016/j.jag.2024.103791
Received 31 December 2023; Received in revised form 8 February 2024; Accepted 2 March 2024
Available online 23 March 2024

millions of points, resulting in exceedingly high computational and memory costs. Recently, similar to superpixel in image processing, there have been many superpoint-based methods (Landrieu and Simonovsky, 2018; Sun et al., 2022; Robert et al., 2023) proposed for clustering semantically homogeneous points into the same group. They are able to describe in detail the relationship between adjacent objects. However, as a pre-processing step before the deep learning network, these methods fail to perform dynamic sampling and clustering for hierarchically extracted semantic features at different stages in the network, limiting their performance.

To address the aforementioned issues, we propose a novel hierarchical point cloud representation framework for 3D semantic segmentation. It combines both dynamic clustering-based Local Feature Aggregating (LFA) blocks and Transformer-based Global Feature Learning (GFL) blocks. We introduce novel semantic feature-based dynamic sampling and clustering methods to LFA blocks, focusing on the local semantic homogeneity of point clusters belonging to any particular object and improving algorithm efficiency. For GFL blocks, we use the dual-attention Transformer to capture long-range context dependencies.

The main contributions of our work can be summarized as follows:

- We design a Transformer-based hierarchical 3D representation framework (named DCTNet) for point cloud segmentation, where the encoder–decoder architecture is highly efficient in capturing local–global information due to its dynamic token generation mechanism in Local Feature Aggregation (LFA) blocks and dual-attention mechanism in Transformer-based Global Feature Learning (GFL) blocks.
- In the encoder, we propose the novel Semantic feature-based Dynamic Sampling (SDS) and Clustering (SDC) methods for dynamic token generation and local feature aggregation. The proposed approaches can not only better identify local semantic homogeneity of 3D objects for improved semantic segmentation, but also greatly improve the computational efficiency compared to traditional sampling and grouping approaches.
- In the decoder, we proposed an efficient semantic feature-guided upsampling method, ensuring a simple yet highly accurate upsampling operation.

The remainder of our paper is organized as follows. Section 2 reviews existing point cloud segmentation methods and summarizes the limitations. Section 3 shows the details of DCTNet. Section 4 presents and discusses the experimental results. Section 5 concludes the paper.

2. Related work

In this section, we first review existing Convolutional Neural Network (CNN)- and Transformer-based point cloud segmentation methods. Then we summarize their limitations and highlight the main contributions of our method.

2.1. Point CNNs

Point CNN-based segmentation methods perform convolution operations directly on the points. Inspired by 2D CNN in image processing, PCNN (Atzmon et al., 2018) defined the convolution of functions over point clouds. It performed the volumetric convolution to arbitrary point clouds by proposing extension and restriction operators. SpiderCNN (Xu et al., 2018) aimed to capture both local and global geometric relationships within point clouds, by employing a set of parameterized convolutional filters. The filters were designed as a product of a simple step function that captures local geodesic information and a Taylor polynomial that ensures expressiveness. PointCNN (Li et al., 2018) proposed a novel χ-transformation strategy to dynamically align local point neighborhoods. It facilitated subsequent convolutional operations, allowing the network to better capture local features. PointConv (Wu et al., 2019) dynamically adjusted its receptive field based on the local point distribution, which allowed the network to effectively capture multi-scale features within irregularly sampled point clouds. Extensive experiments of PointConv demonstrated its robustness to variations in point densities and complex geometric structures. RS-CNN (Liu et al., 2019b) focused on utilizing geometric topology relationships among points to optimize convolutional weights, which allowed the network to obtain discriminative shape awareness.

Besides, the Graph Convolution Network (GCN) is also widely used in point cloud segmentation. It performs convolution operations on points connected with a graph structure, which is beneficial to the local feature aggregation. Based on PointNet (Charles et al., 2017), DGCNN (Wang et al., 2019b) proposed dynamic edge convolution (named EdgeConv) operated on local neighborhoods. EdgeConv captured edge-wise relationships between points, enabling the network to understand local structures and their variations. ResMRGCN (Li et al., 2023a) combined both the advantages of CNNs and GCNs, introducing the residual/dense connections and dilated convolutions of CNNs to the GCN architecture. It made GCNs deeper and proved the positive effect of such a combination. Despite the great success achieved by point CNNs, it is still challenging for them to capture long-range dependencies and global context efficiently. Point CNNs typically operate in local neighborhoods, and while hierarchical structures can help capture some global features, they might not handle contextual relationships as effectively as Transformers.

2.2. Point transformers

The application of Transformers in 3D point cloud segmentation has achieved great success. They can be broadly categorized into two main groups: global Transformer-based methods and local Transformer-based methods.

Global Point Transformers. The global Transformer methods (Guo et al., 2021; Hui et al., 2021; Zhang et al., 2022a; Sun et al., 2023; Robert et al., 2023; Guo et al., 2023; Li et al., 2023c) focus on learning long-range dependency relationships across the entire point cloud, which is the most straightforward in 3D Transformer designing. Point Cloud Transformer (PCT) (Guo et al., 2021) is a representative work. It fed neighborhood-embedded points into a series of stacked Transformer blocks for global feature learning. The main drawback of the global Transformer is the high computational cost, which is caused by its $\mathcal{O}(N^2D)$ complexity, where N is the number of input points, and D is the feature dimension. The runtime of global Transformer methods grows quadratically as the number of input points grows (Liu et al., 2023). Therefore, it is challenging for global Transformer methods to process large-scale scenes for remote sensing. Recently, there have been many efficient global Transformer methods (Hui et al., 2021; Zhang et al., 2022a; Sun et al., 2023; Robert et al., 2023) proposed for point cloud segmentation. PPT-Net (Hui et al., 2021) proposed a hierarchical encoder–decoder framework to reduce the number of points gradually. Instead of using a pure Transformer architecture, it introduced graph convolution-based (Wang et al., 2019b) embedding for local feature aggregation, which not only enhances long-term dependencies of the point clouds but also reduces the computational cost. PatchFromer (Zhang et al., 2022a), SPFormer (Sun et al., 2023), and SPT (Robert et al., 2023) all used Transformer blocks to capture global features from aggregated superpoint-based local features. However, the static point clustering strategy they used cannot adaptively serve the semantic features extracted at different stages of the network, limiting their performance.

Local Point Transformers. The local Transformer methods (Zhao et al., 2021b; Lai et al., 2022; Gao et al., 2022; Liu et al., 2023) focus on extracting local information on a group of subsets of the target point cloud, which could be generally divided into two categories: neighborhood-based and window-based strategy. Point Transformer (Zhao et al., 2021b) is a representative work of neighborhood-based local Transformers. It has a hierarchical encoder–decoder framework, applying the Transformer blocks to the neighborhood conducted

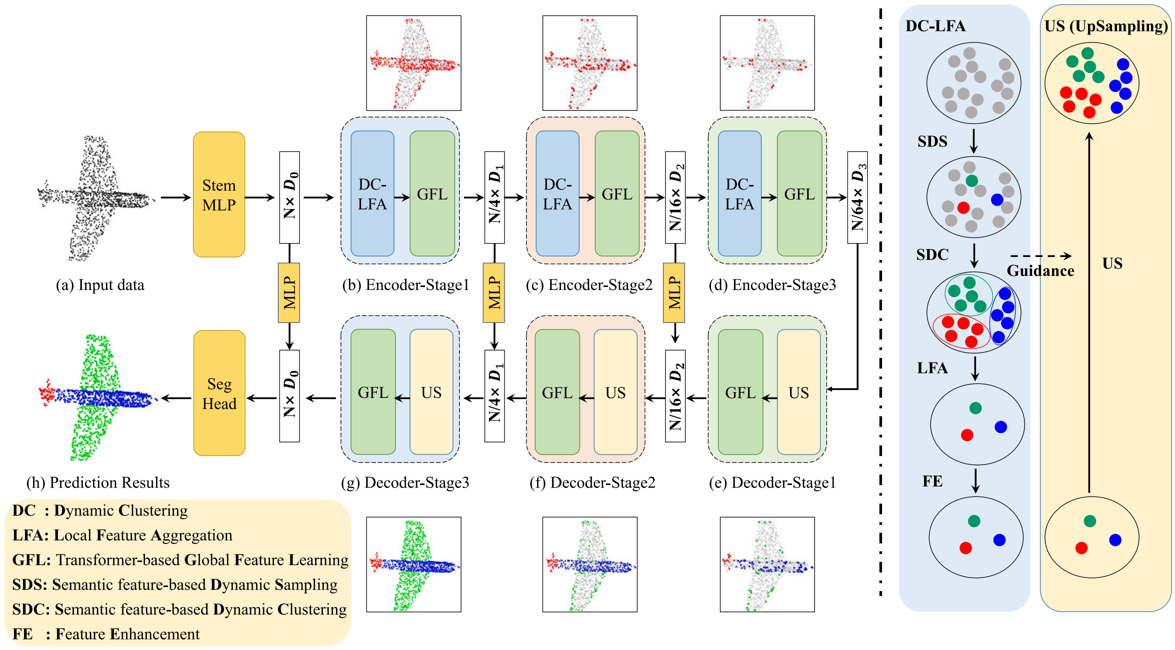


Fig. 1. Hierarchical encoder–decoder structure of DCTNet for point cloud segmentation. Dynamic clustering-based LFA blocks and Transformer-based GFL blocks are designed for local–global feature representation. The airplane model is taken as an example to illustrate the details of the method. For the six subfigures corresponding to six stages, the top three show the hierarchically sampled points in the encoder, and the bottom three show the semantic information of upsampling points in the decoder.

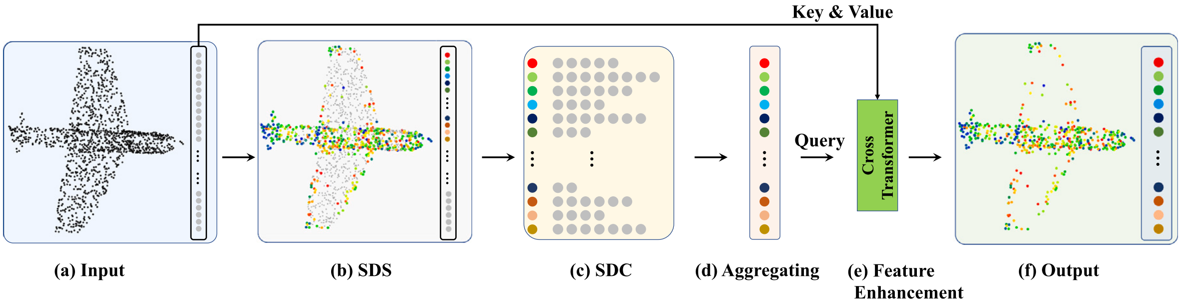


Fig. 2. Pipeline of the dynamic clustering-based LFA block. Semantic feature-based Dynamic Sampling (SDS) and Clustering (SDC) are proposed to ensure local semantic homogeneity in clusters, facilitating local feature aggregating. The non-gray points in (b) are the selected sampling points. (c) shows that we build a cluster for each sampling point. Due to the semantic feature-based dynamic clustering, the number of points in each cluster is different.

by kNN searching for each sampling point. FlatFormer (Liu et al., 2023) used Transformer blocks to extract window-based local features and designed a window shift strategy to achieve global feature learning. However, such indirect global feature learning methods limit the performance of 3D Transformers in long-range context dependency modeling.

To address the aforementioned drawbacks of existing 3D Transformers in point cloud segmentation, we propose a novel hierarchical Transformer-based point cloud representation framework, DCTNet. Different from static point clustering, the proposed dynamic sampling and clustering method not only adaptively aggregates local features to ensure homogeneity in local neighborhoods, but also greatly reduces the computational costs. Given the dynamically aggregated local features, we use the dual-attention Transformer to capture global features. Finally, instead of using trilinear interpolation for point upsampling in the decoder, we propose an efficient semantic feature-guided upsampling method, ensuring a simple yet highly accurate upsampling operation.

3. Dynamic clustering transformer network

This section shows the details of DCTNet. We first present the overall pipeline, then introduce its main blocks: dynamic clustering-based LFA block, Transformer-based GFL block, and semantic feature-guided upsampling block.

3.1. Overview

The overall pipeline of DCTNet is shown in Fig. 1, taking the airplane model as an example to illustrate the details of the method. The original point cloud with/without normal is taken as input to the encoder. Firstly, a stem MLP block (Qian et al., 2022) is designed to project the input data into a higher-dimension space. Secondly, the projected features are fed into several stages in a hierarchical manner for local and global feature extraction. Each stage in the encoder consists of two blocks: a dynamic clustering-based LFA block and a Transformer-based GFL block. Thirdly, the extracted features by the aforementioned stages are taken as input to the decoder. Specifically, the decoder follows the U-Net design, symmetric to the encoder structure described above. As shown in Fig. 1, each stage in the decoder consists of two blocks: a semantic feature-guided upsampling block and a Transformer-based GFL block which is exactly the same as the corresponding block in the encoder. Lastly, an MLP head layer is used to get the final prediction for each point, which consists of two linear layers with batch normalization and ReLU.

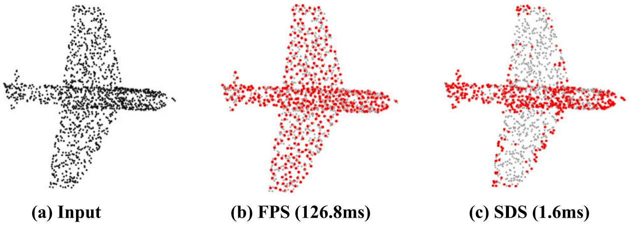


Fig. 3. Comparison of sampling results (downsampling from 2048 to 512 points). Compared with FPS, our SDS focuses on key discriminative geometric areas, retaining fewer points in the flat areas but more points in the nose, tail, and wing contours. Additionally, it is more efficient, nearly 80× faster than FPS.

3.2. Dynamic clustering-based LFA block

The dynamic clustering-based LFA block is designed to achieve discriminative local feature extraction. Our LFA block consists of three key steps: point cloud dynamic clustering, local feature aggregating, and feature enhancement. The pipeline of the LFA block is shown in Fig. 2. The first step is to achieve point cloud sampling and generate semantically homogeneous clusters for sampling points. The second step is to aggregate the point features in the same cluster. The last step is to establish the connection between the aggregated sampling points and input features, enhancing the sampling point features and mitigating feature loss caused by aggregating.

Point Cloud Dynamic Clustering. We propose SDS and SDC methods for point cloud sampling and clustering. For our implementation of SDS, given an input point set $P = \{p_i\}_{i=1}^{N} \in R^{N\times D}$, where D is the dimension of the input feature, we first compute the local density d_i of each point p_i according to its k-nearest neighborhood Φ_i in the feature space:

$$d_i = exp\left(-\frac{1}{k}\sum_{p_j\in\Phi_i}\left\|p_i - p_j\right\|^2\right). \tag{1}$$

According to $P = \{p_i\}_{i=1}^{N}$, we denote $\Gamma = \{d_i\}_{i=1}^{N}$. Secondly, we calculate a distance indicator δ_i for p_i, which can be expressed as:

$$\delta_i = \begin{cases} min_{j:d_j\in\Omega_i}\left\|p_i - p_j\right\|^2, & if\ \Omega_i \neq \emptyset \\ max_{j:d_j\in\Gamma}\left\|p_i - p_j\right\|^2, & otherwise \end{cases} \tag{2}$$

where $\Omega_i = \{d_j \in \Gamma | \forall d_j > d_i\}$. According to this equation, δ_i can be understood as the minimal feature distance between p_i and any other points with higher local density. For the point with the highest local density, its distance indicator is defined as the maximal feature distance between it and any other points. Given d_i and δ_i, we combine them to get the score of each point, which can be expressed as $\delta_i \times d_i$. A higher score means this point has a more representative feature and then is more suitable to be selected as the sampling point. Therefore, according to the sampling rate, we choose the points with the highest scores as sampling points. Based on semantic features, the sampling points are dynamically updated in each stage of the network. The computational complexity of SDS is $\mathcal{O}(N^2D)$. As shown in Fig. 3, compared with FPS, our SDS retains fewer points in the flat areas but more points in the key areas, such as the nose, tail, and wing contours of the airplane. This could provide more useful information for network learning. Additionally, it is more efficient, about 80× faster than FPS.

Thirdly, given the sampling point set $\mathbb{S} = \{s_i\}_{i=1}^{S} \in R^{S\times D}$, we design the SDC method to construct a cluster for each s_i. Specifically, in the feature space, small feature distances mean similar semantic information. Therefore, we assign every point in P to the nearest sampling point in $\mathbb{S}$ based on the feature distances. As such, each s_i has a cluster C_i with local semantic homogeneity, facilitating local feature aggregating. According to dynamically generated sampling points, the clustering process is also dynamically updated in each stage based on semantic features. As shown in Fig. 4, for points at the boundaries of fuselage and wings, kNN grouping tends to include points from two different parts into the same group, which may disturb the local feature aggregating. However, our SDC method is able to cluster points with similar semantic information, ensuring local semantic homogeneity within the same group. The computational complexity of SDC is $\mathcal{O}(NS)$.

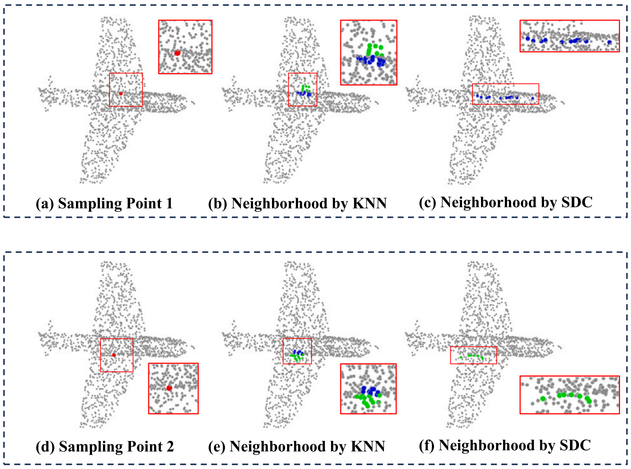


Fig. 4. Grouping results of different methods. Compared with kNN, our SDC is able to cluster points with similar semantic information, ensuring local semantic homogeneity within the same group. The fuselage and wings are colored blue and green respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Local Feature Aggregating. Given clusters $\mathbb{C} = \{C_i\}_{i=1}^{S}$, we utilize a learning-based weighted average algorithm to achieve the local feature aggregating. Since the cluster points in C_i have similar semantic features, an intuitive method is to average them directly, obtaining the aggregated s_i with local information, which can be expressed as:

$$s_i = \underset{j\in C_i}{\text{average}}\,(C_{ij}), \tag{3}$$

where C_{ij} denotes the jth cluster point in C_i. However, it is still hard for points in the same cluster to have the same importance for the network. This simple average may lead to information loss. Therefore, we implement a learnable attention score set $A = \{a_i\}_{i=1}^{N}$ for all points in P. Then for the cluster C_i, the aggregated s_i can be expressed as:

$$s_i = \frac{\sum_{j\in C_i}\exp(a_j)C_{ij}}{\sum_{j\in C_i}\exp(a_j)}, \tag{4}$$

where a_j is the learnable attention score of the point C_{ij}. As such, the aggregated s_i is able to describe the local semantic information more accurately. As such, the aggregated sampling point set $\mathbb{S} = \{s_i\}_{i=1}^{S}$ is obtained. The relationship between the sampling points and cluster points is also stored for the point cloud upsampling in the decoder.

Feature Enhancement. Given the sampling point set $\mathbb{S}$, we design a cross-attention Transformer to establish the connection between $\mathbb{S}$ and input features P, enhancing sampling point features and mitigating information loss caused by the aggregating process.

Specifically, as shown in Fig. 2, we first generate $Query$ matrix based on $\mathbb{S}$, and $Key, Value$ matrices based on P:

$$\begin{aligned} Q &= \mathbb{S}W_Q, \\ K &= PW_K, \\ V &= PW_V, \end{aligned} \tag{5}$$

where Q, K, V denote $Query$, Key, and $Value$ matrices. W_Q, W_K, W_V are learnable weight matrices. Then, the attention map M can be formulated as:

$$M = softmax(\frac{QK^T}{\sqrt{D}} + A). \tag{6}$$

The size of QK^T is $S \times N$, where each element $m_{i,j} \in R^{S\times N}$ represents the feature similarity between ith sampling point in $\mathbb{S}$ and jth input point in P. Additionally, A is defined in the process of Local Feature Aggregating, representing the importance of each point for network decision-making. In implementation, the size of A is $1 \times N$, which is inconsistent with QK^T. Therefore, for our implementation, we repeat A along rows, extending its size to $S\times N$. The element addition between QK^T and A means that both feature similarity and point importance are considered in our cross-attention Transformer. Finally, the enhanced sampling point set $\mathbb{S}$ can be generated by multiplying M and V, with the size of $S \times D$.

3.3. Transformer-based GFL block

We use the Transformer to achieve global feature learning, thanks to its remarkable ability of long-range context dependency modeling. The dual-attention Transformer (Han et al., 2021; Lu et al., 2022) has been proven more effective in global feature learning than vanilla point-wise or channel-wise Transformers. Therefore, we use the dual-attention Transformer in our GFL block. The Point-wise Self-Attention (PSA) in the dual-attention Transformer is used to build the spatial relationship between points, achieving long-range context dependency modeling. Similarly, the Channel-wise Self-Attention (CSA) in the dual-attention Transformer is used to explore the difference between feature channels, highlighting the role of interaction across various channels (Han et al., 2021). By combining these two kinds of self-attention mechanisms, our GFL block is able to capture global features from multiple perspectives.

PSA and CSA have similar algorithm flows. Specifically, taking the sampling point set $\mathbb{S}$ as input, we first project it into three different feature spaces to generate $Query$, Key, and $Value$ matrices:

$$\begin{aligned} Query &= \mathbb{S}W_Q, \\ Key &= \mathbb{S}W_K, \\ Value &= \mathbb{S}W_V, \end{aligned} \tag{7}$$

where W_Q, W_K, W_V are learnable weight matrices. Secondly, for the PSA, the attention map $M_P \in R^{S\times S}$ can be formulated as:

$$M_P = softmax(\frac{QK^T}{\sqrt{D}} + B), \tag{8}$$

where Q, K denote the $Query$, Key matrices, and B is a learnable position encoding matrix defined by Zhao et al. (2021b). M_P and $Value$ matrices are multiplied to generate the new feature map F_P as the output of PSA, of the same size as $\mathbb{S}$. Thirdly, for the CSA, the attention map $M_C \in R^{D\times D}$ can be formulated as:

$$M_C = softmax(\frac{K^TQ}{\sqrt{D}}). \tag{9}$$

$Value$ and M_C matrices are multiplied to generate the new feature map F_C as the output of CSA, of the same size as $\mathbb{S}$.

Given global feature maps F_P and F_C, we combine them by the element-wise addition:

$$F_G = F_P + F_C. \tag{10}$$

Lastly, we apply a skip connection between F_G and the input feature set $\mathbb{S}$:

$$F_G = \mathbb{S} + LBR(F_G), \tag{11}$$

where F_G is the final global feature map, and LBR denotes the combination of $Linear$, $BatchNorm$, and $ReLU$.

Table 1
The network configurations of DCTNet on the ShapeNet part segmentation dataset, including the network depth, sampling rate (denoted as Sr), neighborhood size in SDS (denoted as k), and network width.

Network depth	Sr	k	Network width
Encoder			
Stage-1	4 ↓	16	128
Stage-2	4 ↓	16	256
Stage-3	4 ↓	16	512
Decoder			
Stage-1	4 ↑	–	256
Stage-2	4 ↑	–	128
Stage-3	4 ↑	–	128

3.4. Semantic feature-guided upsampling block

As shown in Fig. 1, since the relationship between sampling points and cluster points has been stored in the encoder, point cloud upsampling can be easily achieved by assigning the semantic features of sampling points to corresponding cluster points. Since the relationship is obtained by semantic feature-based clustering, the upsampling process is named semantic feature-guided upsampling.

As such, compared with commonly used point cloud interpolation methods (Qi et al., 2017; Zhao et al., 2021b; Hui et al., 2021), the efficiency of our semantic feature-guided upsampling process is improved while ensuring that the semantic features of upsampling points are not easily smoothed out.

4. Experiments

In this section, we first present the implementation details of our method, including hardware configuration, training strategy, and hyperparameter settings. Secondly, we present the performance of our network on two public segmentation datasets (ShapeNet (Wu et al., 2015) and Airborne MultiSpectral LiDAR (MS-LiDAR) dataset (Zhao et al., 2021a), which are synthetic and real-scanned datasets respectively). We also compared our method with SOTA works in point cloud segmentation. Lastly, we conducted ablation studies to verify the effectiveness of each main component in our framework.

4.1. Implementation details

The specific network configurations of DCTNet on different datasets are shown in Tables 1 and 2. We implemented DCTNet with PyTorch and trained it on an NVIDIA GeForce RTX 3090 GPU. The network was trained with the SGD Optimizer, with a momentum of 0.9 and weight decay of 0.0001. The initial learning rate was set to 0.001, with a cosine annealing schedule to adjust the learning rate at every epoch. The network was trained for 250 epochs. The batch size was set as 16 for the ShapeNet part segmentation dataset (Wu et al., 2015), and 8 for the airborne MultiSpectral LiDAR (MS-LiDAR) dataset (Zhao et al., 2021a).

4.2. Part segmentation

Datasets and Metrics. The ShapeNet dataset contains 16872 synthetic models with 16 shape categories. They were split into 13998 samples for training and 2874 samples for testing, following Point Transformer (Zhao et al., 2021b). This dataset has 50 part labels, and each object has at least two parts. For a fair comparison, each input point cloud was downsampled to 2048 points. For the evaluation metrics, we used the instance-wise mean Intersection over Union (referred to as mIoU in this paper) and Frame Per Second to measure the accuracy and inference efficiency of algorithms respectively.

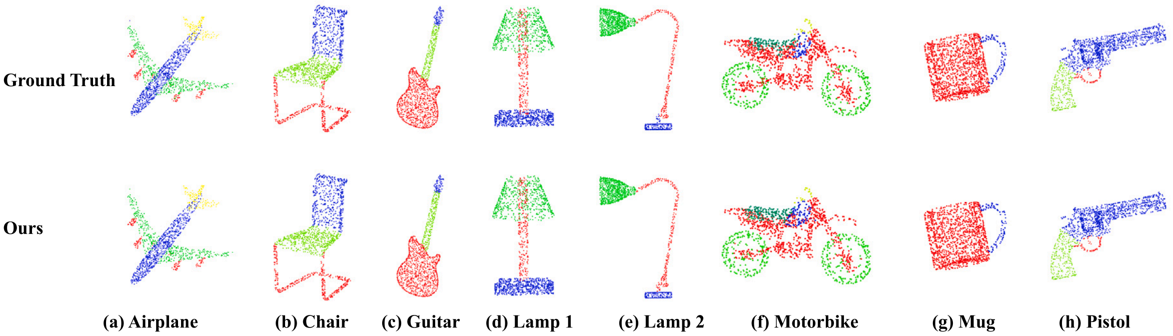


Fig. 5. Part segmentation results from the ShapeNet dataset. As can be seen, our segmentation predictions are faithful to ground truth.

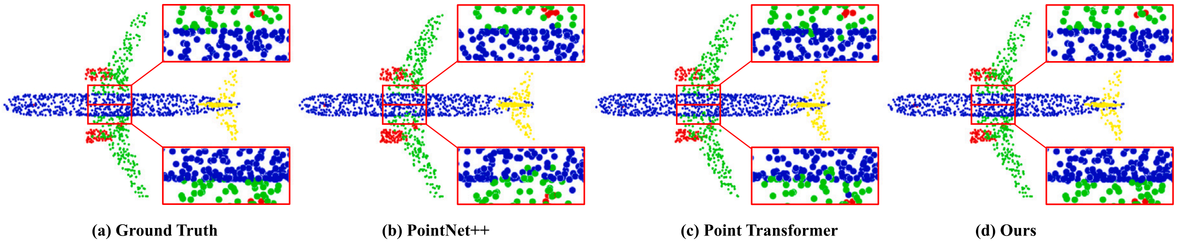


Fig. 6. Airplane segmentation results from different methods on the ShapeNet dataset. Our method achieves the best results at the boundaries of adjacent parts. The fuselage, wings, engines, and tail are colored blue, green, red, and yellow respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
The network configurations of DCTNet on the airborne MS-LiDAR dataset, including the network depth, sampling rate (denoted as Sr), neighborhood size in SDS (denoted as k), and network width.

Network depth	Sr	k	Network width
Encoder			
Stage-1	4 ↓	16	128
Stage-2	4 ↓	16	256
Stage-3	4 ↓	16	512
Stage-4	4 ↓	16	1024
Decoder			
Stage-1	4 ↑	–	512
Stage-2	4 ↑	–	256
Stage-3	4 ↑	–	128
Stage-4	4 ↑	–	128

Table 3
Part segmentation results on the ShapeNet dataset.

Methods	mIoU (%)	Frame Per Sec.
PointNet++ (Qi et al., 2017)	85.1	6.8
PCNN (Atzmon et al., 2018)	85.1	–
SpiderCNN (Xu et al., 2018)	85.3	13.9
SGPN (Wang et al., 2018)	85.8	–
DGCNN (Wang et al., 2019b)	85.2	19.7
PointConv (Wu et al., 2019)	85.7	15.9
RS-CNN (Liu et al., 2019b)	86.2	9.4
InterpCNN (Mao et al., 2019)	86.3	–
DensePoint (Liu et al., 2019a)	86.4	–
PCT (Guo et al., 2021)	86.4	–
PT (Zhao et al., 2021b)	**86.6**	7.8
PVT (Zhang et al., 2022b)	86.5	9.7
ST (Lai et al., 2022)	**86.6**	2.2
PatchFormer (Zhang et al., 2022a)	86.5	27.8
APES (Wu et al., 2023)	85.8	–
Li et al. (2023b)	86.0	–
Hassan et al. (2023)	86.3	–
Ours	**86.6**	**37.0**

Performance Comparison. We compared our DCTNet with SOTA segmentation methods. As shown in Table 3, DCTNet achieves the competitive mIoU (86.6%) with existing SOTA methods. As for the inference efficiency, compared with those algorithms (Point Transformer (Zhao et al., 2021b), Stratified Transformer (Lai et al., 2022), etc.) that used FPS for downsampling, our method achieves a higher Frame Per Second (37.0). These results indicate our DCTNet greatly improves processing efficiency while maintaining accuracy. Several visual results of part segmentation are shown in Fig. 5. As can be seen, segmentation predictions from DCTNet are faithful to the ground truth. Moreover, Fig. 6 shows the airplane segmentation results of different methods. DCTNet achieves the best segmentation at the boundaries of adjacent parts (fuselage and wings).

4.3. Airborne MS-LiDAR segmentation

Datasets and Metrics. Most recently, a large-scale airborne MultiSpectral LiDAR (MS-LiDAR) dataset was introduced in Zhao et al. (2021a). We tested DCTNet on this dataset to explore its performance in practical remote sensing applications. The MS-LiDAR dataset was captured by a Teledyne Optech Titan MS-LiDAR system (Zhao et al., 2021a). In addition to three-dimensional coordinates, each point also has three channels with wavelengths of 1,550 nm (MIR), 1,064 nm (NIR), and 532 nm (Green). There are six categories in the dataset: Road, Building, Grass, Tree, Soil, and Powerline. The dataset was divided into 13 subsets, where subsets 1-10 were taken as training data, while subsets 11-13 were taken as testing data. For fair comparison, we took the same data pre-processing (data fusion, normalization, and training/testing sample generation) methods described in Zhao et al. (2021a). Each subset is partitioned to a series of local blocks as training/test samples, and each of them contains 4096 points with six channels. We used Overall Accuracy (OA), mIoU, and average F_1 score for performance evaluation, and provided the F_1 score for each category.

Performance Comparison. The semantic segmentation results of airborne MS-LiDAR data are shown in Table 4, in the form of a confusion matrix. Our DCTNet achieves excellent F_1 scores of over 85% for all categories except soil. Most of the misclassification points are found

Table 4
Confusion matrix (%) of DCTNet on the airborne MS-LiDAR dataset. The numbers in the last three rows represent the precision, recall, and F_1 score for each class.

Categories	True label						
		Road	Building	Grass	Tree	Soil	Powerline
Prediction label	Road	85.9	3.0	0.0	0.1	9.9	0.0
	Building	11.5	92.9	0.8	0.6	32.8	0.0
	Grass	0.2	0.9	98.8	1.3	0.2	26.8
	Tree	0.1	0.7	0.3	97.9	0.9	0.1
	Soil	2.4	2.6	0.0	0.1	56.2	0.0
	Powerline	0.0	0.0	0.0	0.0	0.0	73.1
Precision		87.8	90.0	99.2	92.2	68.2	93.3
Recall		85.9	92.9	98.8	97.9	56.2	73.1
F_1		86.8	91.4	99.0	95.0	61.6	82.0

Table 5
Quantitative comparison (%) of semantic segmentation performance on the airborne MS-LiDAR dataset. The highest evaluation score is shown in bold type.

Methods	Average F_1 score	mIoU	OA	Frame Per Sec.
PointNet++ (Qi et al., 2017)	72.1	58.6	90.1	3.1
DGCNN (Wang et al., 2019b)	71.6	51.0	91.4	11.6
RSCNN (Liu et al., 2019b)	73.9	56.1	91.0	6.3
GACNet (Wang et al., 2019a)	67.7	51.0	90.0	3.6
AGConv (Zhou et al., 2021)	76.9	71.2	93.3	3.2
SE-PointNet++ (Jing et al., 2021)	75.9	60.2	91.2	–
FR-GCNet (Zhao et al., 2021a)	78.6	65.8	93.6	–
PT (Zhao et al., 2021b)	80.5	73.6	93.1	3.5
PPT-Net (Hui et al., 2021)	80.1	73.6	92.7	23.1
Xiao et al. (Xiao et al., 2022)	83.3	79.3	94.0	–
PatchFormer (Zhang et al., 2022a)	82.4	77.8	93.1	15.9
ResMRGCN-28 (Li et al., 2023a)	81.1	74.0	93.3	21.9
Ours	**86.0**	**80.2**	**95.0**	**23.3**

in areas of soil and roads. This is because the geometric characteristics of the soil are very similar to roads, which tends to confuse the network. More feature discrimination approaches are planned in our future work to improve the ability of the model to distinguish the soil.

The comparison results are shown in Table 5. Our DCTNet outperforms all benchmarked methods, achieving the best results in terms of both OA (95.0%), average F_1 score (86.0%), and mIoU (80.2%). In terms of inference efficiency, we also surpass the efficient Transformer methods such as PPT-Net (Hui et al., 2021) and PatchFormer (Zhang et al., 2022a). Visualization of comparison results are shown in Fig. 7. These results show that our DCTNet has an excellent performance in MS-LiDAR point cloud segmentation, exceeding that of previous methods.

4.4. Network ablation

Ablation studies were conducted on the airborne MS-LiDAR dataset, to verify the effectiveness of key blocks in DCTNet.

Dynamic Clustering-based LFA Block. In the LFA block (Section 3.2) of DCTNet, we propose SDS and SDC methods for local feature aggregating, then use the cross-attention Transformer for feature enhancement.

To evaluate their effectiveness, we first replaced the SDS method with FPS. As shown in Table 6 (Row 2), the DCTNet network with FPS obtains a lower average F_1 score (85.2%) than with the SDS method (86.0%). Additionally, FPS is also very time-consuming, which reduces the inference efficiency of the network. As shown in Table 6 (Row 2), we can see the DCTNet with FPS has a much lower Frame Per Second[1] (2.7) than the original one (23.3).

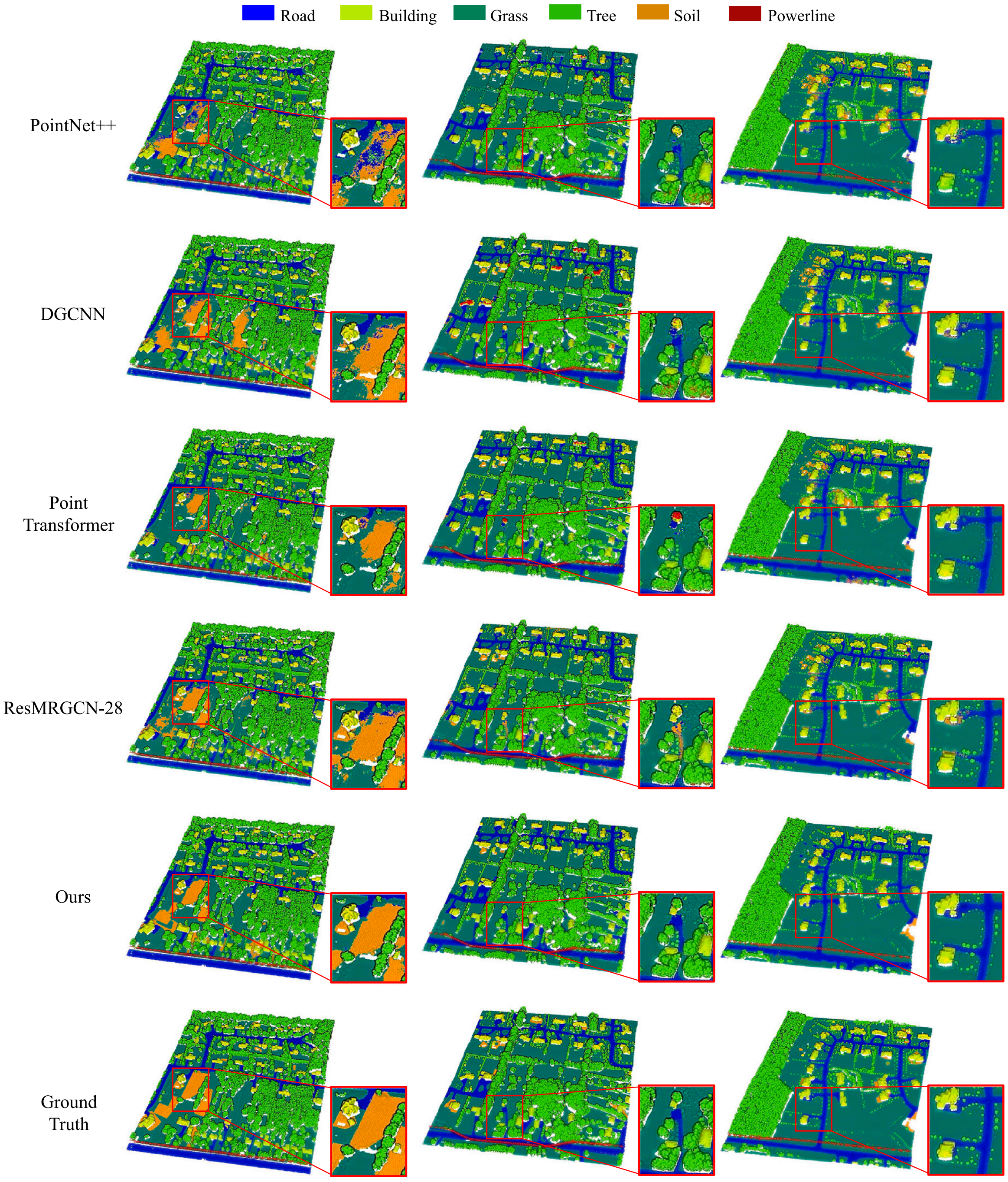
Secondly, we used the "k-Nearest Neighborhood (kNN) + MultiLayer Perceptron (MLP) + Maxpooling" to replace the proposed SDC method, following the local feature extraction process in PointNet++ (Qi et al., 2017). Correspondingly, the point cloud upsampling method in the decoder was also replaced with the trilinear-interpolation upsampling. As shown in Table 6 (Row 3), after replacement, the average F_1 score of the network is reduced to 85.0% from 86.0%, which highlights the importance of the SDC method. The main reason is that the kNN method achieves feature grouping only based on three-dimensional coordinates, ignoring local semantic homogeneity. In terms of inference efficiency, Table 6 (Row 3) shows that the Frame Per Second (16.5) of the network after the replacement is also lower than the original one (23.3). These results demonstrate that the proposed SDC method is able to improve local feature aggregation.

Thirdly, we also conducted detailed ablation studies for the learnable score set A defined in Section 3.2. As shown in Table 6 (Row 4 and 5), we firstly remove A from both Eqs. (4) and (6), which actually makes our local feature aggregation process degenerate to Eq. (3). Although the points in the same cluster have similar semantic features, it is still hard for them to have the same importance for the network. This simple average may lead to information loss. The drops in all three metrics in Table 6 (Row 4) have confirmed it. Besides, we also explored to only remove A from the cross-attention Transformer layer of feature enhancement which is defined in Eq. (6). We also observed slight drops in all metrics in Table 6 (Row 5), which demonstrates that the learnable score set A is beneficial to local feature enhancement.

Finally, we removed the cross-attention Transformer in the LFA block. As shown in Table 6 (Row 6), without the cross-attention Transformer, the average F_1 score is reduced from 86.0% to 84.7%.

Dual-attention Transformer-based GFL Block. Dual-attention Transformers (Section 3.3) have been proven effective by previous works (Han et al., 2021; Lu et al., 2022). We conducted ablation studies to verify that dual-attention Transformers outperformed vanilla Transformers with only point-wise or channel-wise self-attention mechanisms. As shown in Table 6 (Row 7), when the channel-wise self-attention is removed, the average F_1 score of DCTNet drops from 86.0% to 85.4%. Similarly, when the point-wise self-attention is removed (

[1] Given that we partitioned the large-scale airborne MS-LiDAR dataset into a group of local samples for processing, the Frame Per Second metric here actually means Sample Per Second.



(a) Test area-11 (b) Test area-12 (c) Test area-13

Fig. 7. Comparison results from different methods on the airborne MS-LiDAR dataset.

Table 6
Ablation study results (%) of key blocks in DCTNet.

Ablation		Average F_1	mIoU	OA	Frame Per Sec.
LFA	SDS →FPS	85.2	78.7	93.6	2.7
	SDC → kNN + MLP	85.0	78.1	94.2	16.5
	$-A$ in both Eq. (4) and Eq. (6)	85.5	78.0	94.1	23.5
	$-A$ in Eq. (6)	85.7	78.9	94.5	23.3
	− Cross-attention Transformer	84.7	77.3	92.7	29.0
GFL	− CSA	85.4	79.2	93.9	24.9
	− PSA	83.4	76.1	91.8	**33.0**
Upsampling	Trilinear interpolation	85.5	77.9	94.0	20.8
	Nearest neighbor interpolation	85.7	79.2	94.2	22.5
DCTNet		**86.0**	**80.2**	**95.0**	23.3

Table 7
Ablation study results (%) for various neighborhood sizes.

k	Airborne MS-LiDAR dataset			ShapeNet dataset
	Average F_1	mIoU	OA	mIoU
4	83.1	78.3	90.9	85.8
8	85.5	79.8	94.4	86.6
16	**86.0**	**80.2**	**95.0**	**86.6**
32	85.9	80.0	94.5	86.3

Table 6, Row 8), there is a similar drop in terms of average F_1 score (from 86.0% to 83.4%).

Point Cloud Upsampling Block. We compared the proposed semantic feature-guided upsampling (Section 3.4) in the decoder with several commonly used upsampling methods, such as trilinear interpolation and nearest neighbor interpolation. The results are shown in Table 6 (Row 9, 10). According to the results, our semantic feature-guided upsampling method outperforms the aforementioned two interpolation methods in terms of all three metrics. This is because our upsampling method assigns the features of the sampling points to the corresponding cluster points, according to the relationship stored in the encoder. This ensures that the semantic features of upsampled points are not easily smoothed out and slightly improves upsampling efficiency.

Sensitivities of Parameters. Since we conduct k-nearest neighborhood for each input point in the process of SDS to compute the local density, we explored the impact of the neighbor point number k on the segmentation performance of DCTNet. A series of k values are selected: 4, 8, 16, 32. In the case of taking both the airborne MS-LiDAR point clouds with 4096 points and ShapeNet point clouds with 2048 points as input, the segmentation performance of DCTNet with different k values is shown in Table 7. From the results, we can see that DCTNet achieves similar results on both datasets in terms of the ablation studies on k. Specifically, the model performance is close when k equals 8, 16, and 32, which means that our model is robust to the neighborhood size k, with the best performance when k is set to 16. However, the performance drops significantly when k equals 4. This may be because the 4-size neighborhood is too small to accurately represent the local density of the point. Overall, the values of k greater than 1/512 of input points work well in our experiments. These ablation studies could help guide users to set appropriate parameters for local density computation for unknown datasets.

5. Conclusion

In this paper, we propose DCTNet, a novel Transformer-based 3D point cloud processing framework that is highly suited for segmenting LiDAR-remote sensing point cloud scenes, as well as general-purpose point cloud segmentation. DCTNet has a hierarchical encoder–decoder structure. For local feature learning, we propose the new Semantic feature-based Dynamic Sampling and Clustering algorithms, acronymed as SDS and SDC respectively. Compared with prevalent sampling and grouping methods, our SDS and SDC are more suitable for semantic information learning, while also facilitating the point cloud upsampling process. For global feature learning, we utilize dual-attention Transformer blocks which excel at modeling long-range dependencies. Our decoder is symmetric to the encoder but contains our newly designed semantic feature-guided upsampling method which improves efficiency and ensures that the semantic features of upsampling points are not easily smoothed out. To our knowledge, DCTNet is the first work to introduce semantic information-based dynamic clustering into 3D Transformers. Extensive experiments on the ShapeNet (Wu et al., 2015) and Airborne MS-LiDAR datasets (Zhao et al., 2021a) demonstrate that DCTNet outperforms previous methods. In terms of algorithm efficiency, the inference speed of DCTNet is 3.8–16.8× faster than existing SOTA models on the ShapeNet dataset, while achieving a competitive mIoU top score of 86.6%. These results show that DCTNet has achieved State-of-the-Art status in point cloud segmentation.

CRediT authorship contribution statement

Dening Lu: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jun Zhou:** Writing – review & editing, Visualization, Methodology, Investigation, Data curation. **Kyle (Yilin) Gao:** Writing – review & editing, Investigation. **Jing Du:** Writing – review & editing, Investigation. **Linlin Xu:** Writing – review & editing, Supervision, Resources, Methodology, Conceptualization. **Jonathan Li:** Writing – review & editing, Supervision, Resources, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under the Discovery Grant No. RGPIN-2022-03741. The first author was sponsored in part by the Chinese Scholarship Council under Grant No. 202106830030.

References

Antonello, M., Wolf, D., Prankl, J., Ghidoni, S., Menegatti, E., Vincze, M., 2018. Multi-view 3D entangled forest for semantic segmentation and mapping. In: IEEE Int. Conf. Robot. Autom.. IEEE, pp. 1855–1862.

Atzmon, M., Maron, H., Lipman, Y., 2018. Point convolutional neural networks by extension operators. arXiv:1803.10091. URL http://arxiv.org/abs/1803.10091.

Bui, L.K., Glennie, C.L., 2023. Estimation of lidar-based gridded DEM uncertainty with varying terrain roughness and point density. ISPRS J. Photogramm. Remote Sens. 7, 100028.

Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J., 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 652–660. http://dx.doi.org/10.1109/CVPR.2017.16.

Chen, J., Cho, Y.K., 2022. CrackEmbed: Point feature embedding for crack segmentation from disaster site point clouds with anomaly detection. Adv. Eng. Inform. 52, 101550.

Chen, C., Wu, H., Yang, Z., Li, Y., 2023. Adaptive coarse-to-fine clustering and terrain feature-aware-based method for reducing LiDAR terrain point clouds. ISPRS J. Photogramm. Remote Sens. 200, 89–105.

Dai, A., Nießner, M., 2018. 3Dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In: Proc. Eur. Conf. Comput. Vis.. pp. 452–468.

Gao, Y., Liu, X., Li, J., Fang, Z., Jiang, X., Huq, K.M.S., 2022. LFT-net: Local feature transformer network for point clouds analysis. IEEE Trans. Intell. Transp. http://dx.doi.org/10.1109/TITS.2022.3140355.

Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M., 2021. PCT: Point cloud transformer. Comput. Vis. Media. 7 (2), 187–199.

Guo, B., Deng, L., Wang, R., Guo, W., Ng, A.H.M., Bai, W., 2023. MCTNet: Multiscale cross-attention-based transformer network for semantic segmentation of large-scale point cloud. IEEE IEEE Trans. Geosci. Remote Sens. 61, 1–20. http://dx.doi.org/10.1109/TGRS.2023.3322579.

Han, X.F., Jin, Y.F., Cheng, H.X., Xiao, G.Q., 2021. Dual transformer for point cloud analysis. arXiv:2104.13044. URL http://arxiv.org/abs/2104.13044.

Hassan, R., Fraz, M., Rajput, A., Shahzad, M., 2023. Residual learning with annularly convolutional neural networks for classification and segmentation of 3D point clouds. Neurocomputing 526, 96–108.

Hui, L., Yang, H., Cheng, M., Xie, J., Yang, J., 2021. Pyramid point cloud transformer for large-scale place recognition. In: Proc. IEEE Int. Conf. Comput. Vis.. pp. 6098–6107.

Jing, Z., Guan, H., Zhao, P., Li, D., Yu, Y., Zang, Y., Wang, H., Li, J., 2021. Multispectral LiDAR point cloud classification using SE-PointNet++. Remote Sens. 13 (13), 2516.

Kundu, A., Yin, X., Fathi, A., Ross, D., Brewington, B., Funkhouser, T., Pantofaru, C., 2020. Virtual multi-view fusion for 3D semantic segmentation. In: Proc. Eur. Conf. Comput. Vis.. Springer, pp. 518–535.

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J., 2022. Stratified transformer for 3D point cloud segmentation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 8500–8509.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 4558–4567.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. PointCNN: Convolution on X-transformed points. In: Proc. Adv. Neural Inf. Process. Syst., Vol. 31. pp. 820–830.

Li, G., Müller, M., Qian, G., Delgadillo, I.C., Abualshour, A., Thabet, A., Ghanem, B., 2023a. DeepGCNs: Making GCNs go as deep as CNNs. IEEE Trans. Pattern Anal. Mach. Intell. 45 (6), 6923–6939. http://dx.doi.org/10.1109/TPAMI.2021.3074057.

Li, Y., Wang, Y., Liu, Y., 2023b. 3D point cloud segmentation based on context feature for sheet metal part boundary recognition. Trans. Instrum. Meas..

Li, J., Wu, H., Xiao, Z., Lu, H., 2022. 3D modeling of laser-scanned trees based on skeleton refined extraction. Int. J. Appl. Earth Obs. Geoinf. 112, 102943.

Li, J., Zhang, Z., Sun, H., Xie, S., Zou, J., Ji, C., Lu, Y., Ren, X., Wang, L., 2023c. GL-net: Semantic segmentation for point clouds of shield tunnel via global feature learning and local feature discriminative aggregation. ISPRS J. Photogramm. Remote Sens. 199, 335–349.

Lin, Y.C., Habib, A., 2022. Semantic segmentation of bridge components and road infrastructure from mobile LiDAR data. ISPRS J. Photogramm. Remote Sens. 6, 100023.

Liu, Y., Fan, B., Meng, G., Lu, J., Xiang, S., Pan, C., 2019a. Densepoint: Learning densely contextual representation for efficient point cloud processing. In: Proc. IEEE Int. Conf. Comput. Vis.. pp. 5239–5248.

Liu, Y., Fan, B., Xiang, S., Pan, C., 2019b. Relation-shape convolutional neural network for point cloud analysis. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 8895–8904.

Liu, Z., Yang, X., Tang, H., Yang, S., Han, S., 2023. FlatFormer: Flattened window attention for efficient point cloud transformer. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 1200–1211.

Lu, D., Gao, K., Xie, Q., Xu, L., Li, J., 2022. 3DPCT: 3D point cloud transformer with dual self-attention. arXiv:2209.11255. URL http://arxiv.org/abs/2209.11255.

Mao, J., Wang, X., Li, H., 2019. Interpolated convolutional networks for 3D point cloud understanding. In: Proc. IEEE Int. Conf. Comput. Vis.. pp. 1578–1587.

Mascaro, R., Teixeira, L., Chli, M., 2021. Diffuser: Multi-view 2d-to-3d label diffusion for semantic scene segmentation. In: IEEE Int. Conf. Robot. Autom.. IEEE, pp. 13589–13595.

Maturana, D., Scherer, S., Sep. 2015. Voxnet: A 3D convolutional neural network for real-time object recognition. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.. pp. 922–928. http://dx.doi.org/10.1109/IROS.2015.7353481.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Proc. Adv. Neural Inf. Process. Syst.. pp. 5099–5108.

Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H.A.A.K., Elhoseiny, M., Ghanem, B., 2022. PointNeXt: Revisiting PointNet++ with improved training and scaling strategies. arXiv:2206.04670. URL http://arxiv.org/abs/2206.04670.

Qin, N., Tan, W., Guan, H., Wang, L., Ma, L., Tao, P., Fatholahi, S., Hu, X., Li, J., 2023. Towards intelligent ground filtering of large-scale topographic point clouds: A comprehensive survey. Int. J. Appl. Earth Obs. Geoinf. 125, 103566.

Riegler, G., O. Ulusoy, A., Geiger, A., 2017. OctNet: Learning deep 3D representations at high resolutions. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 3577–3586. http://dx.doi.org/10.1109/CVPR.2017.701.

Robert, D., Raguet, H., Landrieu, L., 2023. Efficient 3D semantic segmentation with superpoint transformer. arXiv:2306.08045. URL http://arxiv.org/abs/2306.08045.

Robert, D., Vallet, B., Landrieu, L., 2022. Learning multi-view aggregation in the wild for large-scale 3D semantic segmentation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 5575–5584.

Sun, J., Qing, C., Tan, J., Xu, X., 2022. Superpoint transformer for 3D scene instance segmentation. arXiv:2211.15766. URL http://arxiv.org/abs/2211.15766.

Sun, J., Qing, C., Tan, J., Xu, X., 2023. Superpoint transformer for 3D scene instance segmentation. In: AAAI Conf. Artif. Intell., Vol. 37, No. 2. pp. 2393–2401.

Tao, P., Tan, K., Ke, T., Liu, S., Zhang, W., Yang, J., Zhu, X., 2022. Recognition of ecological vegetation fairy circles in intertidal salt marshes from UAV LiDAR point clouds. Int. J. Appl. Earth Obs. Geoinf. 114, 103029.

Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J., 2019. Kpconv: Flexible and deformable convolution for point clouds. In: Proc. IEEE Int. Conf. Comput. Vis.. pp. 6411–6420.

Wang, L., Huang, Y., Hou, Y., Zhang, S., Shan, J., 2019a. Graph attention convolution for point cloud semantic segmentation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 10296–10305.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019b. Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. 38 (5), 1–12. http://dx.doi.org/10.1145/3326362.

Wang, W., Yu, R., Huang, Q., Neumann, U., 2018. SGPN: similarity group proposal network for 3D point cloud instance segmentation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 2569–2578.

Wei, R., Ye, C., Ge, Y., Li, Y., Li, J., 2023. Dynamic graph attention networks for point cloud landslide segmentation. Int. J. Appl. Earth Obs. Geoinf. 124, 103542.

Wu, W., Qi, Z., Fuxin, L., 2019. PointConv: Deep convolutional networks on 3D point clouds. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 9621–9630. http://dx.doi.org/10.1109/CVPR.2019.00985.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D shapenets: A deep representation for volumetric shapes. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 1912–1920.

Wu, C., Zheng, J., Pfrommer, J., Beyerer, J., 2023. Attention-based point cloud edge sampling. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 5333–5343.

Xiao, W., Cao, H., Tang, M., Zhang, Z., Chen, N., 2023. 3D urban object change detection from aerial and terrestrial point clouds: A review. Int. J. Appl. Earth Obs. Geoinf. 118, 103258.

Xiao, K., Qian, J., Li, T., Peng, Y., 2022. Multispectral LiDAR point cloud segmentation for land cover leveraging semantic fusion in deep learning network. Remote Sens. 15 (1), 243.

Xu, Y., Fan, T., Xu, M., Zeng, L., Qiao, Y., 2018. SpiderCNN: Deep learning on point sets with parameterized convolutional filters. In: Proc. Eur. Conf. Comput. Vis., Vol. 11212. pp. 90–105.

Zhang, C., Wan, H., Shen, X., Wu, Z., 2022a. Patchformer: An efficient point transformer with patch attention. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 11799–11808.

Zhang, C., Wan, H., Shen, X., Wu, Z., 2022b. PVT: Point-voxel transformer for point cloud learning. Int. J. Intell. Syst. 37 (12), 11985–12008.

Zhao, P., Guan, H., Li, D., Yu, Y., Wang, H., Gao, K., Junior, J.M., Li, J., 2021a. Airborne multispectral LiDAR point cloud classification with a feature reasoning-based graph convolution network. Int. J. Appl. Earth Obs. Geoinf. 105, 102634.

Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V., 2021b. Point transformer. In: Proc. IEEE Int. Conf. Comput. Vis.. pp. 16259–16268.

Zhou, H., Feng, Y., Fang, M., Wei, M., Qin, J., Lu, T., 2021. Adaptive graph convolution for point cloud analysis. In: Proc. IEEE Int. Conf. Comput. Vis.. pp. 4965–4974.

Zhou, Y., Tuzel, O., 2018. Voxelnet: End-to-end learning for point cloud based 3D object detection. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. pp. 4490–4499.

Zováthi, Ö., Nagy, B., Benedek, C., 2022. Point cloud registration and change detection in urban environment using an onboard lidar sensor and MLS reference data. Int. J. Appl. Earth Obs. Geoinf. 110, 102767.