TopoSeg: Topology-aware Segmentation for Point Clouds

Weiquan Liu[1*] , **Hanyun Guo**[1*] , **Weini Zhang**[1] , **Yu Zang**[1†] , **Cheng Wang**[1] , **Jonathan Li**[2]
[1]Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics,
Xiamen University, Xiamen, China
[2]Departments of Geography and Environmental Management and Systems Design Engineering,
University of Waterloo, Waterloo, Canada
wqliu@xmu.edu.cn, hyguo@stu.xmu.edu.cn, weini_zhang@163.com, zangyu7@126.com,
cwang@xmu.edu.cn, junli@uwaterloo.ca

Abstract

Point cloud segmentation plays an important role in AI applications such as autonomous driving, AR, and VR. However, previous point cloud segmentation neural networks rarely pay attention to the topological correctness of the segmentation results. In this paper, focusing on the perspective of topology awareness. First, to optimize the distribution of segmented predictions from the perspective of topology, we introduce the persistent homology theory in topology into a 3D point cloud deep learning framework. Second, we propose a topology-aware 3D point cloud segmentation module, TopoSeg. Specifically, we design a topological loss function embedded in TopoSeg module, which imposes topological constraints on the segmentation of 3D point clouds. Experiments show that our proposed TopoSeg module can be easily embedded into the point cloud segmentation network and improve the segmentation performance. In addition, based on the constructed topology loss function, we propose a topology-aware point cloud edge extraction algorithm, which is demonstrated that has strong robustness.

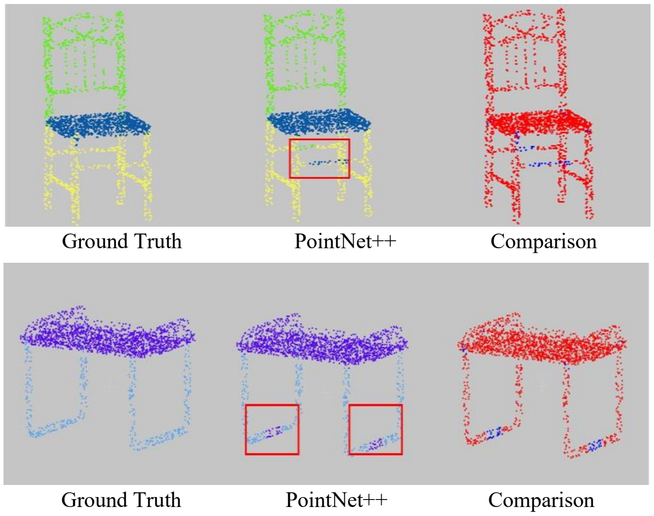


Figure 1: Typical topological errors (red bounding boxes) in the segmentation results of PointNet++. The first, second, third column are the ground truth, the segmentation results by PointNet++, the comparison between segmentation results and ground truth, respectively. The red points indicate the correctly segmented point, and blue points indicate the wrongly segmented point.

1 Introduction

With the development of sensor technology and the rapid growth of the amount of point cloud data, 3D point clouds have been widely used in many Artificial Intelligence (AI) fields, such as autonomous driving, indoor navigation, Augmented Reality (AR), virtual reality (VR), etc. In these applications, point cloud segmentation is a fundamental and important task, which has received a lot of attention.

Since PointNet [Qi *et al.*, 2017a] proposed the first neural network directly operating on unordered point sets, many effective deep learning networks for point cloud segmentation have emerged. These networks mainly use the cross-entropy loss as the loss function. However, although these networks integrate local information when extracting features, the cross-entropy loss still considers each point independently without global topology constraints, which may lead to some topological errors in the segmentation results, as shown in Figure 1.

In recent years, Topological Data Analysis (TDA), a field combining topology theory and data analysis, has developed rapidly. TDA provides a set of tools to effectively capture the topological information of high-dimensional data space and better describe the shape of the data. Therefore, TDA has been successfully applied in many fields such as point cloud processing [Brüel-Gabrielsson *et al.*, 2020], biomedical analysis [Offroy and Duponchel, 2016], complex network analysis [Taylor *et al.*, 2015]. The combination of TDA and deep learning can introduce topological information into the neural networks for training, that provides a new perspective for exploring the intrinsic characteristics of the data.

In this paper, we introduce explicit topological constraints into the neural networks for point clouds to refine segmentation results by using Persistent Homology (PH) in topology. Subsequently, we propose a topology-aware 3D point

[*] indicates equal contribution

[†] Corresponding author

cloud segmentation neural network module, named TopoSeg, to constrain the global topology of the point cloud segmentation results. In addition, we design a topological loss function embedded into the TopoSeg module, so that the deep neural network and TDA promote each other.

Specifically, using deep neural network alone is difficult to capture the topological information of the point cloud. The addition of topological loss enables the network to obtain the segmentation results with the correct topology while capturing the geometric features of the point clouds. Thus, we use the output of the neural networks as the observed data for TDA makes the original data rich in semantic information, which helps to better capture the relationships of more complex data.

The experiments show that the proposed TopoSeg module and topological loss effectively reduce the topological errors and improve the performance of baseline networks. Besides, we apply the proposed topology-aware point cloud segmentation network to the edge point detection task. The results show that using topological loss improves the accuracy of edge point detection, and the detected edge points have more reasonable topological structures, which provide a better candidate point set as basis for the following line segment fitting.

The main contributions of our work are as follows:

- We propose a topology-aware 3D point cloud segmentation neural network module, TopoSeg, to constrain the global topology of the point cloud segmentation results.
- We construct a topological loss function, embedded into the TopoSeg module, for point cloud segmentation based on persistent homology, so that making the segmentation results more reasonable in topological structures.
- We design a strategy to embed the proposed TopoSeg module and topological loss function into the existing point cloud segmentation networks for end-to-end training, which makes the segmentation results have the similar topological structures with the ground truth.
- We apply the proposed TopoSeg module to the edge point detection task, which improves the accuracy of edge point detection.

2 Related Work

2.1 Persistent Homology and Machine Learning

Topological Data Analysis (TDA) is a rapidly developing field combining data science and topology theory, that provides a set of powerful tools to measure the intrinsic shape of data. In the field of TDA, Persistence Homology (PH) is a well-established method to track the changes of topological features across multiple scales and find more persistent topological patterns of the underlying data space. Due to the differentiable properties, topological information can be integrated into machine and deep learning methods through PH to improve performance. According to the application of PH in machine and deep learning, previous work can be roughly divided into two categories: feature-engineering-based methods and topology-loss-based methods.

Feature-engineering-based Methods

In feature-engineering-based methods, the topological information obtained by PH is integrated into machine and deep learning models as fixed predefined features. This kind of methods tackle classification or distance calculation tasks according to topological features. However, the distribution of the input data can not be adjusted to approximate some specific topological structure.

[Hofer *et al.*, 2017] proposed a CNN-based classifier with topological signatures extracted from image, shape or graph as input. This method exploited a novel topological input layer learning a parameterized projection of topological information as feature descriptors, and improved the performance of 2D object shapes and social network graphs classification tasks. [Xie *et al.*, 2014] proposed a fast method for 3D shape segmentation and labeling via extreme learning machine, which reduced the training time by approximately two orders of magnitude, both for face-level and super-face-level. [Zhang *et al.*, 2020] proposed a fusion-aware 3D point convolution which operates directly over the progressively acquired and online reconstructed scene surface. This method used geodesic distance to capture the underlying geometry and topology of 3D surfaces, and achieved online segmentation at close-to-interactive frame-rate. Other methods [Carrière *et al.*, 2015; Adams *et al.*, 2017] proposed some strategies to vectorize persistence diagrams, which can be used in kernel based classifiers. These strategies were integrated into a general neural network framework for graph classification [Carriere *et al.*, 2019]. In addition, the topological features extracted by PH can be used for deep learning interpretability [Gabrielsson and Carlsson, 2019], adversarial attacks [Gebhart and Schrater, 2017], automated architecture design [Carlsson and Gabrielsson, 2020] and complexity measures [Guss and Salakhutdinov, 2018; Rieck *et al.*, 2018] for neural networks.

Topology-loss-based Methods

The differentiability of PH makes it possible to optimize the data distribution via topology loss function. This kind of methods take the parameters and outputs of neural networks as input to compute PH, and allow gradient-based optimization algorithms to push the topology of input to the desired structure.

[Hofer *et al.*, 2019] applied PH to the latent vectors learned from the encoder, and designed a differentiable topological loss term to promote the extracted features to have certain topological structures or connectivity properties. In previous works [Hu *et al.*, 2019; Clough *et al.*, 2020], the topology loss functions were designed for end-to-end neural networks to guide the image semantic segmentation results to have specified topological structures. [Gabrielsson *et al.*, 2020] used PH on the weights of neural networks for regularization, allowing the weights to tend to form a small number of clusters.

3 Method

Similar to other baseline networks, our point cloud segmentation network uses cross-entropy loss to achieve per-point label accuracy. On this basis, we embed a topological loss term

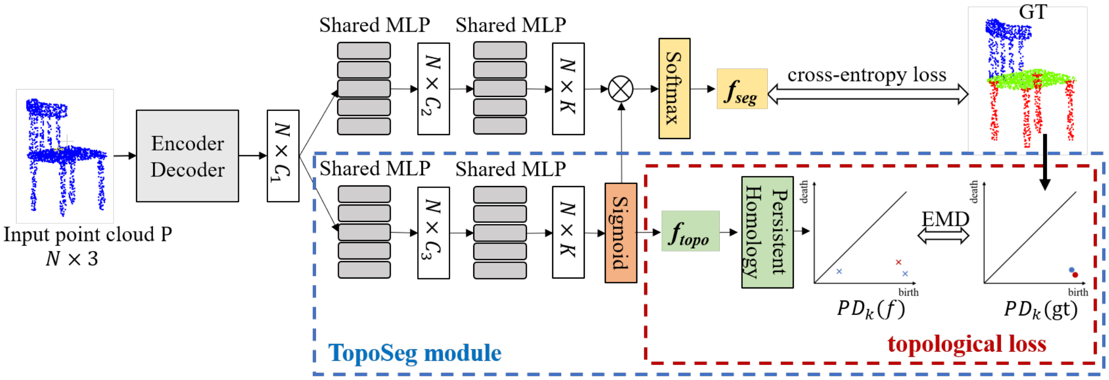


Figure 2: Point cloud segmentation network architecture embedded with the proposed TopoSeg module and topological loss. The upper branch is the Point cloud segmentation network, and the bottom branch is the TopoSeg module.

based on PH into the proposed TopoSeg module to refine the topology of the segmentation results, as shown in Figure 2.

3.1 Topological Loss

Persistent Homology

The study object of PH is an increasing nested sequence of simplicial complexes, called filtration. An abstract simplcial complex $\mathbb{K}$ is a finite collection of simplices (i.e., k-dimensional polytopes like points, lines, triangles, tetrahedron) which is closed under taking subsets. Defining a real-valued monotonic function $f : \mathcal{K} \to \mathbb{R}$ on a simplicial complex $\mathcal{K}$ and the level sub-complexes $\mathcal{K}(\alpha) = f^{-1}(-\infty, \alpha]$, we can obtain the super-level set filtration of $\mathcal{K}$ by decreasing the parameter α,

$$\begin{aligned} \emptyset \subset \mathcal{K}(\alpha_0) \subset \mathcal{K}(\alpha_1) \subset \cdots \subset \mathcal{K}(\alpha_L) = \mathcal{K}, \\ \alpha_0 > \alpha_1 > \cdots > \alpha_L. \end{aligned} \tag{1}$$

PH studies the topology of the underlying space by tracking the appearance and disappearance of topological features at different scales. The importance of topological features is reflected by the length of their lifetime. The persistent topological features which survive on a wide range of scales are considered to be stable features, while the short-lived features are considered to be caused by noise or specific parameter selection. PH describes the lifetime of a topological feature by the birth and death time (scale parameter) concisely. There are many summary representations of PH, among which persistence diagram is a common and popular topological signature. A persistence diagram $PD_k(f) \subset \mathbb{R}^2$ is a multiset of k-dimensional features' (birth, death) tuples, which can be regarded as a mapping from filtration to a point set,

$$PD_k : (\mathcal{K}, f) \to \{b_i, d_i\}, \tag{2}$$

where each point $\{b_i, d_i\} \in PD_k$ corresponds to a topological feature that born at b_i and died at d_i.

The advantage of using PH to analyze the topology is that it can better reflect more stable topological characteristics of data space and is robust to certain data perturbations.

Topological Loss Definition

In our method, we use super-level set filtration based on a fixed simplicial complex to construct topological loss. For simplicity, we first consider binary classification task for a certain category. For a point cloud, which is defined as a finite points in Euclidean space, the Vietoris-Rips complex is a natural choice to approximate the topology of the underlying space for homology computation. Given a set of points in n-dimensional Euclidean space $P = \{p_0, p_1, \ldots, p_m \in \mathbb{R}^n\}$ and a fixed radius scale ϵ, the Vietoris–Rips complex $\mathcal{VR}_\epsilon(P)$ contains simplices in which the distances of any pair of points are less than ϵ. The formal definition of the Vietoris–Rips complex is as below,

$$\mathcal{VR}_\epsilon(P) = \{\sigma \subseteq \mathcal{P}(P) | d(p_i, p_j) \leq \epsilon, \forall p_i, p_j \in \sigma\}, \tag{3}$$

where $\mathcal{P}(P)$ is the power set of P, and $d(\cdot, \cdot)$ is the Euclidean distance.

Then we define a function on the simplicial complex $\mathcal{VR}_\epsilon(P)$ for filtration $f : \mathcal{VR}_\epsilon(P) \to R$, where the function value at each point $f(p_i)$ is the prediction obtained by the neural network, and the function value of each simplex σ is the smallest function value of each point in the simplex. Thus we build a mapping from the simplex σ to the point $w(\sigma) = argmin_{p \in \sigma} f(p)$. We construct a super-level set filtration and calculate the persistence diagrams $PD_k(f)$ and $PD_k(g)$ corresponding to the network predictions and the ground truth respectively. Notice that as the filter value decreases, the existing topological features will be merged and die, which means that all points in the persistence diagram $PD_k(\cdot)$ are below the diagonal of the first quadrant. Because the function value is non-negative, the death time of topological features that survive to the end can be recorded as 0.

In particular, for the ground truth, the function value at each point is either 0 or 1. All meaningful topological features are born at $f = 1$ and die at $f = 0$, thus all persistence points in $PD_k(g)$ are at (1,0). As mentioned before, it is usually considered that the long-lived features represent the real and stable topology of the data, that is, the persistence points far away from the diagonal are corresponding to the stable topological features and contain more crucial information.

Since $PD_k(f)$ and $PD_k(g)$ are point sets in $\mathbb{R}^2$, the Earth Mover's Distance (EMD) is a appropriate metric to measure the similarity of two point sets. The EMD between the point set S_1 and S_2 is shown as below

$$EMD(S_1, S_2) = \min_{\Phi:S_1 \to S_2} \sum_{x \in S_1} ||x - \Phi(x)||^2, \tag{4}$$

where $\Phi : S_1 \to S_2$ is a bijection. To calculate the EMD between two point sets, we need to find an optimal match between S_1 and S_2. Because the number of points in $PD_k(f)$ and $PD_k(g)$, denoted as $|PD_k(f)|, |PD_k(g)|$, may be different, we design the following matching strategy to calculate the EMD:

- If $|PD_k(f)| > |PD_k(g)|$, then the top-$|PD_k(g)|$ points in $PD_k(f)$ closest to $(1, 0)$ are matched with $(1, 0)$, and the remaining unmatched points in $PD_k(f)$ are matched to the diagonal.
- If $|PD_k(f)| <= |PD_k(g)|$, then all points in $PD_k(f)$ are matched with $(1, 0)$, the remaining unmatched points in $PD_k(g)$ are matched to the diagonal.

This matching strategy is more efficient than the optimal matching algorithm of two arbitrary point sets. It only needs to sort the distances from the points in $PD_k(f)$ to $(0, 1)$, and the time complexity is $O(nlogn)$. Thus, for $i - th$ category, the topological loss L_{topo-i} is as follows

$$\begin{aligned} L_{topo-i}(f_i, g_i) &= \sum_k EMD(PD_k(f_i), PD_k(g_i)) \\ &= \sum_k \sum_{p \in PD_k(f_i)} ||p - \phi_k^*(p)||^2. \end{aligned} \tag{5}$$

For the multi-class segmentation task, it can be regarded as multiple binary classification problems and the total topological loss L_{topo} can be the average and sum of the loss L_{topo_i} for each category. Thus, the total loss function of the network is defined as follows

$$\begin{aligned} &L(f, g) = L_{ce}(f_{seg}, g) + \lambda L_{topo}(f_{topo}, g) \\ &L_{topo}(f_{topo}, g) = \frac{1}{C} \sum_{i=1}^{C} L_{topo-i}(f_{topo-i}, g_i), \end{aligned} \tag{6}$$

where g denotes the ground truth, f_{seg}, f_{topo} are the output of the network (as shown in Figure 2). f_{seg} is the output for multi-class segmentation task, where each dimension of the output corresponding to the probability of a category is mutually exclusive, while f_{topo} is the binary classification output for all C categories, which is non-exclusive. In Figure 2, Softmax makes the sum of the outputs of all classes equal to 1, while Sigmoid calculates the output of each class separately, i.e., the output value of one class does not affect the output value of another, which is non-exclusive. The definition of f_{topo} is non-exclusive refers to the non-exclusive characteristic of Sigmoid. L_{ce}, L_{topo} denotes cross-entropy loss and topological loss respectively, λ is the weight of the topological loss.

It should be noted that the point cloud segmentation network is still mainly constrained by the cross-entropy loss. The topological loss cannot work alone, which is mainly to refine a relatively reasonable probability mapping to get a better segmentation result with more precise topological structures, rather than infer topology directly from unreasonable probability output.

Topology Loss Optimization

To introduce topological loss into the deep learning framework for optimization, we need to calculate the gradient of the topological loss. From Eq.(5), it can be viewed that points in persistence diagrams that participate in the loss calculation depend on some specific function values (thresholds), at which the topological features appear or disappear. According to the chain rule, we need to define a mapping from the points in the persistence diagram $PD_k(f)$ to the points in the original point cloud. From the calculation of PH, it can be seen that each persistent point in $PD_k(f)$ represents the threshold corresponding to a topological feature's birth and death. And according to the definition of the function $f : \mathcal{VR}_\epsilon(P) \to R$, the function value is exactly the output of the neural network at points in original point cloud. Considering the construction of the super-level set filtration, when the threshold drops to α_i, some new points $\{p \in P | f(p) = \alpha_i\}$ will join in the construction of the level sub-complexes. This will lead to the birth of new topological features or the death of original topological features, and the persistence diagram will record these events in the form of $(birth, death)$ tuples. Thus we define a mapping h_k,

$$h_k : \{b_i, d_i\} \to (cp(\sigma_b), cp(\sigma_d)), \tag{7}$$

where $\{b_i, d_i\}$ is the persistence point in $PD_k(f)$, σ_b and σ_d are the key simplexes leading to the birth and death of the corresponding topological feature, respectively. $cp(\sigma)$ is the key point in original point cloud to determine the value of the simplex σ. Here, we use the super-level set filtration, and the corresponding $cp(\sigma)$ is the point with the smallest function value in the simplex σ.

In this way, we establish the relationship between the persistence points and the key points in the original point cloud. Then we calculate the gradient of topological loss at each persistence point $\{b_i, d_i\} \in PD_k(f)$, and the gradients are assigned to the corresponding key points $cp(\sigma_b), cp(\sigma_d)$ during backpropagation. The gradient of topological loss (Eq.(5)) is defined as follows,

$$\begin{aligned} &\nabla_w L_{topo-i}(f_i, g_i) = \sum_k \sum_{p \in PD_k(f_i)} 2[p - \phi_k^*(p)] \\ &= \sum_k \sum_{p \in PD_k(f_i)} 2[f(cb(p)) - birth(\phi_k^*(p))] \frac{\partial f(cb(p))}{\partial w} \\ &+ 2[f(cd(p)) - death(\phi_k^*(p))] \frac{\partial f(cd(p))}{\partial w}, \end{aligned} \tag{8}$$

where $cb(p)$ and $cd(p)$ represent the birth and death key points in original point cloud corresponding to the persistence point p. The output of the network and the weights of the network are denoted by $f(\cdot)$ and w, respectively.

3.2 Network Architecture

Considering that the main constraint of the network is still the cross-entropy loss, and the topological loss term is used to refine the results from the perspective of topological structure. Thus, as shown in Figure 2, we design a general strategy to incorporate the proposed TopoSeg module and topological

	mean	aero	bag	cap	car	chair	Ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	Skate board	table
PointNet++	84.97	84.40	81.81	70.69	**84.85**	84.85	83.65	**94.27**	86.77	76.29	96.46	**76.38**	75.91	91.75	63.74	**93.39**	**86.73**
PointNet++ + TopoSeg	**85.99**	**85.20**	**82.06**	**83.67**	83.86	**87.26**	**87.79**	93.62	**90.06**	**78.21**	**96.53**	74.92	**79.63**	**91.94**	**65.15**	93.33	86.71
DGCNN	91.33	**89.30**	93.69	90.11	87.53	**93.45**	76.80	**94.98**	88.69	87.53	**97.70**	**82.06**	**98.73**	90.76	**80.40**	**94.99**	91.80
DGCNN + TopoSeg	**91.54**	88.75	**94.12**	**93.35**	**87.92**	93.06	**79.65**	94.91	**91.35**	**89.02**	97.69	80.65	98.67	**92.68**	78.01	94.67	**92.22**
CurveNet	91.27	**88.70**	94.06	88.24	86.11	**93.63**	84.51	95.42	90.87	88.90	97.62	**81.02**	98.77	**94.06**	**78.93**	93.25	91.10
CurveNet + TopoSeg	**91.43**	88.56	**95.59**	**90.91**	**87.27**	93.53	**85.69**	**95.58**	**91.25**	**89.13**	**98.00**	78.09	**99.32**	93.28	78.20	**93.43**	**91.39**

Table 1: Part segmentation results on ShapeNet. Metric is Accuracy(%).

loss into point cloud segmentation networks. This network extracts per point features via the backbone network, and contains two branches. The first branch is the topologically constrained branch, which further extracts features through multiple MLP layers, and uses a sigmoid layer to predict the probabilities of each point belonging to each category. The output of this branch f_{topo} participates in the calculation of topological loss. The second branch applies multiple MLP layers, and the output is multiplied by the weight, which is the output of the first branch. Then a softmax layer is used to obtain per point scores f_{seg} for the calculation of cross-entropy loss.

In summary, the topological loss can be simply incorporated into any existing segmentation networks that provides per point predictions. The topological loss function can help adjust the weights of the neural network, and ultimately make the topological structure of the network output similar to the ground truth.

4 Experiment

Our proposed TopoSeg module can be embedded into many point cloud segmentation backbone networks to improve the topology performance of the outputs. To evaluate the effectiveness of the proposed TopoSeg module and topological loss, we mainly designed two groups of experiments, 3D object part segmentation (Section 4.1) and edge point detection (Section 4.2). In this work, all the experiments are conducted in Linux with a Geforce RTX 3090 GPU. The network is built based on the Pytorch framework, and the Dionysus package is used for the calculation of PH.

4.1 3D Object Part Segmentation

In this part, we aim at the task of 3D object part segmentation. Considering that objects of the same category often have similar components, while the component structures of objects belonging to different categories are more likely to be different, the network can better learn the topology of the components corresponding to a specific object category.

Dataset and Implement Details

This group of experiments are conducted on the large-scale public dataset of 3D shapes, ShapeNet [Chang *et al.*, 2015], which is co-established by the researchers at Princeton, Stanford and TTIC. ShapeNet contains 16,881 3D shapes from 16 main object categories and 50 part categories, and most objects consist of 2-5 parts. We select 512 points by random sampling for each training sample.

As for the PH calculation settings, according to the data distribution, we set the fixed radius scale $\epsilon = 0.05$ to construct the Vietoris–Rips complex $\mathcal{VR}_\epsilon(P)$, and we calculate 0-2 dimensional persistence diagrams. For the sake of computational efficiency, we set the maximum number of persistence points to 300, and ignore the points exceeding the maximum number. Thus the dimension of the diagram for each point cloud is [50, 3, 300, 2].

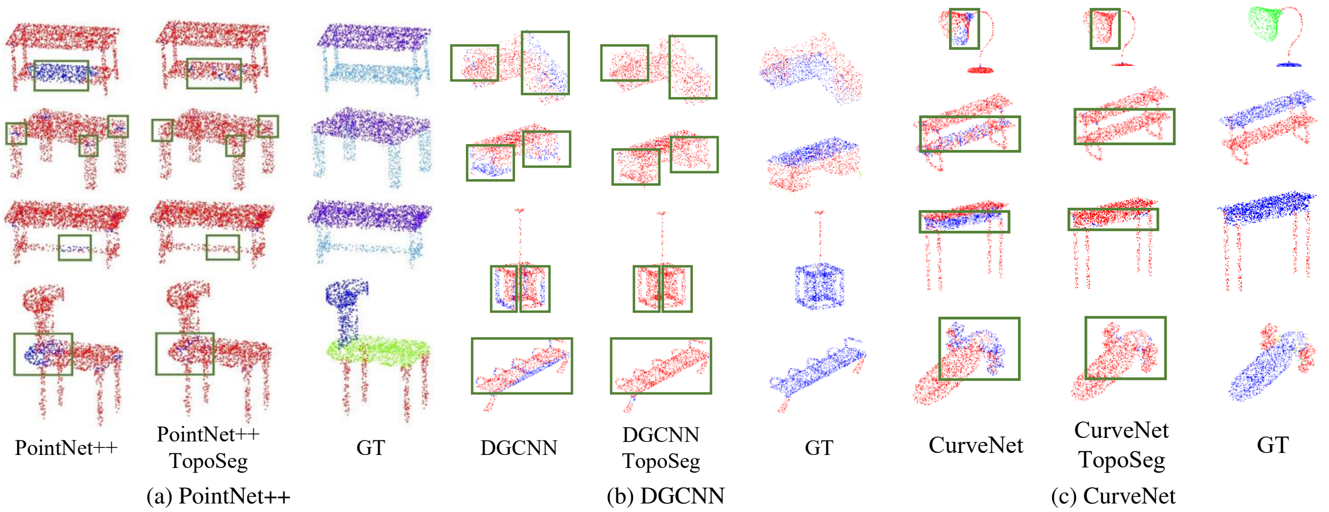
As for the network and training settings, Adam optimizer with initial learning rate of 0.001 is used for training, and the learning rate is reduced by half every 20 epochs. The batch size and total training epochs is set to 64 and 200, respectively. In the loss function Eq.(6), the weight of topology loss λ is set to 0.001.

Comparison Results of 3D Object Part Segmentation

To evaluate the effectiveness of the topological loss, in this paper, we corporate the topological loss into the baseline networks (PointNet++ [Qi *et al.*, 2017b], DGCNN [Wang *et al.*, 2019] and CurveNet [Xiang *et al.*, 2021]) for comparison and show the qualitative and visualization results to illustrate the improvement of our method. We take the accuracy as the evaluation metric. It should be noted that due to the limitation of computing resources, only part of the data is used for training. Therefore, under the same experiment environment, we retrain the original baseline networks with the same settings in the original paper, and compare the results with our method, shown in Table 1. Table 1 demonstrates that when our proposed TopoSeg module is embedded into an existing point cloud segmentation network, the segmentation accuracy of about half of the objects is improved.

In Table 1, the proposed TopoSeg model decreases considerably the accuraccy for class ’motor’. We consider the reasons is that the label of motor is not detailed enough, resulting in a large number of rings or other complex topological structures in the samples, thus leading to some negative effects of our module.

It should be noted that, for one thing, the proposed TopoSeg module aims to improve the segmentation accuracy for the result. On the other side, which is more important, TopoSeg is design to improve the topology of the outputs. However, in the experimental results, we find that both accuracy and mean IoU are hard to evaluate the topological structure. Actually, the results of applying mean IoU and accuracy are similar. Thus, we only list accuracy as the metric in the manuscript. In addition, we find current metrics focus mainly on the segmentation performance, while hard to evaluate the topological structure, maybe that is the reason why the improvement on part segmentation seems to be marginal in Table 1. Thus, we also provide plenty of visualized results to assist evaluate the effectiveness of the proposed approach (Figure 3).



(a) PointNet++ (b) DGCNN (c) CurveNet

Figure 3: Visualization of typical samples on ShapeNet by embedding our proposed TopoSeg module and topological loss. The red points indicate the correctly segmented point, and blue points indicate the wrongly segmented point.

Figure 3 shows the visualization results of some typical samples. To show the comparison results more intuitively, we use red and blue points to indicate correct predictions and wrong predictions, respectively. Each line from left to right are the results of baseline networks, baseline networks with TopoSeg module and the ground truth (GT). As can be seen from Figure 3, after adding the proposed TopoSeg module and topological loss, the segmented components are relatively more complete and some topological errors such as fractures and holes can be corrected.

4.2 Edge Point Detection

For 3D edge line structure extraction, edge-point-based methods are commonly used, which detect potential edge points first and then fitting 3D line segments by least square fitting, region growing etc. Edge-point-based methods extract sharp edges and retain more details, but the performance depends on the accuracy of edge point detection.

In this part, we consider the edge point detection task as the binary-class segmentation task, that is, given a point cloud $P = \{p_i | i = 1, \ldots, N\}$, we assign a edge or not edge label to each point in P.

Dataset and Implement Details

This group of experiments are conducted on the dataset established in [Yu *et al.*, 2018], which contains 24 CAD models and 12 daily object models with sharp edges. The training data are obtained by virtual scanning of CAD models, and we sample around the annotated edge line segments to obtain edge points. Here, the point whose shortest distance from the edge is less than 0.01 is selected as the edge point.

As for the PH calculation settings, the fixed radius scale ϵ is set to 0.01, and we calculate 0-2 dimensional persistence diagrams with up to 300 persistence points. With the network and training settings, we use Adam optimizer with initial learning rate of 0.001, and the learning rate is reduced by half every 10 epochs. The batch size, total training epochs and the weight of topology loss λ are set to 64, 200 and 0.0005, respectively. The segment fitting algorithm used in this paper is the method in the previous work of [Lin *et al.*, 2017], which is a robust line segment grouping method with false alarm filtering.

	accuracy	F1 score
PointNet++ [Qi *et al.*, 2017b]	0.9034	0.6267
PointNet++ + TopoSeg (Ours)	**0.9118**	**0.6386**

Table 2: Results of edge point detection.

Comparison Results of Edge Point Detection

For edge point detection task, we use accuracy and F1 value as evaluation metrics, and we select PointNet++ as baseline to compare with the topology-aware network proposed in this paper. The results are shown in Table 2. It can be seen that the accuracy of the network with topological constraints reach 0.9118, exceeding PointNet++ by 0.84%. It indicates that the proposed topology-aware network performs well in edge point detection task, which provide more reliable candidate points for the next line segment fitting stage.

Figure 4 shows some visualization results of edge point detection. The first column is the input point cloud and ground truth, and the second column is the detected points of PointNet++. In order to show the results more intuitively, we calculate the distance from the detected edge point to the true edges and map it to different colors, shown in the third column, where blue indicates short distance and red indicates long distance. And the fourth and fifth column is the results of our topology aware network. As can be seen from Figure 4, compared with PointNet++ [Qi *et al.*, 2017b], the potential edge points detected by our method are more distributed near the edges, and less distributed on the surface of the object or far away from the edges, which are more accurate in topological structure.

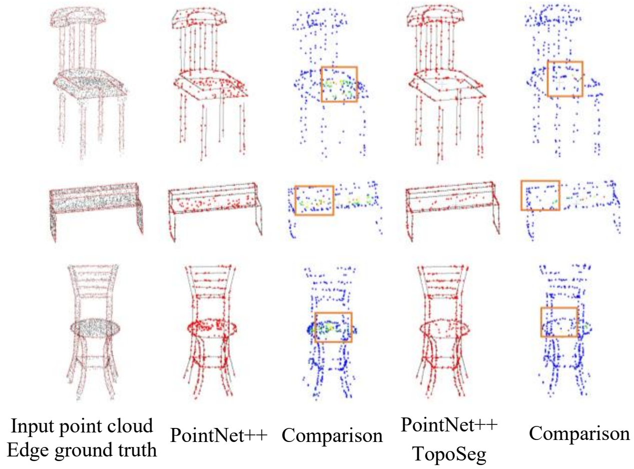


Figure 4: Visualization results of edge point detection.

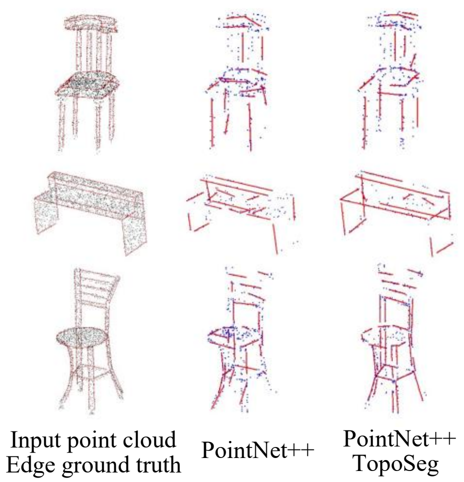


Figure 5: Visualization results of line segment fitting.

The result of line fitting for detected edge points is shown in Figure 5, where from left to right are the input point cloud, the results of line fitting on the edge points extracted by PointNet++ and our network. From the fitting results we can see that the edge point detected by our network are more reasonable and the fitting lines are closer to the real edges. And due to fewer false detected points, there are fewer false lines distributed on the surface of the object in the final fitting results.

5 Discussion

During the training phase, embedding the TopoSeg module will increase the training time by about 2-3 times, compare to the original network. During the testing phase, the effect of the TopoSeg module has been reflected in the parameters of the network through the topology loss. Moreover, it does not participate in the calculation during testing, so it will not bring extra the computational time in the testing phase. For the memory, there is almost no extra cost during both training and testing phases.

Our TopoSeg module is not learning based, but to constrain the network by utilizing the topology information. For the sparse point clouds, they provide less information for the segmentation network to learn. So after embedding our TopoSeg module, the outputs are improved more significantly. While, for dense point clouds, the performance still improve, but not that significantly. Thus, the network indeed gets benefit from the TopoSeg module, while the point sampling will affect improvement degree. Furthermore, for the noised case, as with the vanilla segmentation network, there is a drop in segmentation accuracy, but our outputs are still better.

6 Conclusion

In this paper, we introduce the persistent homology theory in topology into the deep learning frameworks in the form of topological loss, to optimize the distribution of predictions from the perspective of topology. Then, we propose a topology-aware 3D point cloud segmentation neural network module, TopoSeg, to constrain the global topology of the point cloud segmentation results. Besides, we design a general strategy to simply embed the topology loss function into the existing point cloud segmentation networks, to improve the performance of the original networks. We demonstrate the effectiveness of our proposed topology loss by two groups of experiments: 3D object part segmentation and edge point detection. The experiments show that the proposed topological loss can effectively reduce the topological errors and improve the performance of baseline networks.

The limitation of our method is the difficulty of extending to large-scale real scene data. The computational complexity of persistent homology is related to the scale of complex. For large-scale data, the computation of persistent homology is very time-consuming, so we only experiment on simple data for the time being. In the future, we consider simplifying the complex to improve the efficiency of the algorithm.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No.61971363, 62171393, 41971424), China Postdoctoral Science Foundation (No.2021M690094), FuXiaQuan National Independent Innovation Demonstration Zone Collaborative Innovation Platform (No. 3502ZCQXT2021003), National Key R&D Program of China (No.2021YFF0704600), and open fund of PDL (No.20215250113).

References

[Adams *et al.*, 2017] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017.

[Brüel-Gabrielsson *et al.*, 2020] Rickard Brüel-Gabrielsson, Vignesh Ganapathi-Subramanian, Primoz Skraba, and Leonidas J Guibas. Topology-aware surface reconstruction for point clouds. In *Computer Graphics Forum*, volume 39, pages 197–207, 2020.

[Carlsson and Gabrielsson, 2020] Gunnar Carlsson and Rickard Brüel Gabrielsson. Topological approaches to deep learning. In *Topological Data Analysis*, pages 119–146. Springer, 2020.

[Carrière *et al.*, 2015] Mathieu Carrière, Steve Y Oudot, and Maks Ovsjanikov. Stable topological signatures for points on 3d shapes. In *Computer Graphics Forum*, volume 34, pages 1–12, 2015.

[Carriere *et al.*, 2019] Mathieu Carriere, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. A general neural network architecture for persistence diagrams and graph classification. *HAL open science: hal-02105788*, 2019.

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

[Clough *et al.*, 2020] James Clough, Nicholas Byrne, Ilkay Oksuz, Veronika A Zimmer, Julia A Schnabel, and Andrew King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access)*, 2020.

[Gabrielsson and Carlsson, 2019] Rickard Brüel Gabrielsson and Gunnar Carlsson. Exposition and interpretation of the topology of neural networks. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)*, pages 1069–1076, 2019.

[Gabrielsson *et al.*, 2020] Rickard Brüel Gabrielsson, Bradley J Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1553–1563, 2020.

[Gebhart and Schrater, 2017] Thomas Gebhart and Paul Schrater. Adversary detection in neural networks via persistent homology. *arXiv preprint arXiv:1711.10056*, 2017.

[Guss and Salakhutdinov, 2018] William H Guss and Ruslan Salakhutdinov. On characterizing the capacity of neural networks using algebraic topology. *arXiv preprint arXiv:1802.04443*, 2018.

[Hofer *et al.*, 2017] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[Hofer *et al.*, 2019] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Mandar Dixit. Connectivity-optimized representation learning via persistent homology. In *International Conference on Machine Learning (ICML)*, pages 2751–2760, 2019.

[Hu *et al.*, 2019] Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[Lin *et al.*, 2017] Yangbin Lin, Cheng Wang, Bili Chen, Dawei Zai, and Jonathan Li. Facet segmentation-based line segment extraction for large-scale point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 55(9):4839–4854, 2017.

[Offroy and Duponchel, 2016] Marc Offroy and Ludovic Duponchel. Topological data analysis: A promising big data exploration tool in biology, analytical chemistry and physical chemistry. *Analytica chimica acta*, 910:1–11, 2016.

[Qi *et al.*, 2017a] 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.

[Qi *et al.*, 2017b] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[Rieck *et al.*, 2018] Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. *arXiv preprint arXiv:1812.09764*, 2018.

[Taylor *et al.*, 2015] Dane Taylor, Florian Klimm, Heather A Harrington, Miroslav Kramár, Konstantin Mischaikow, Mason A Porter, and Peter J Mucha. Topological data analysis of contagion maps for examining spreading processes on networks. *Nature Communications*, 6(1):1–11, 2015.

[Wang *et al.*, 2019] 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*, 38(5):1–12, 2019.

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

[Xie *et al.*, 2014] Zhige Xie, Kai Xu, Ligang Liu, and Yueshan Xiong. 3d shape segmentation and labeling via extreme learning machine. In *Computer graphics forum*, volume 33, pages 85–95, 2014.

[Yu *et al.*, 2018] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Ec-net: an edge-aware point set consolidation network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 386–402, 2018.

[Zhang *et al.*, 2020] Jiazhao Zhang, Chenyang Zhu, Lintao Zheng, and Kai Xu. Fusion-aware point convolution for online semantic 3d scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4534–4543, 2020.