3D Shape Segmentation via Attentive Nonuniform Downsampling

Zhenyu Shu, Xufei Sun*, Chaoyi Pang, and Shiqing Xin

***Abstract*—The segmentation of 3D shapes is a critical aspect of shape analysis. However, most existing methods for 3D shape segmentation treat each face of the original mesh model with equal importance. This uniform approach becomes problematic in areas where the faces are smaller but denser, especially around the junctions of different segments. In such regions, greater importance should be assigned compared to the flatter areas. To address this issue, this paper proposes a novel 3D shape segmentation method that incorporates attentive nonuniform sampling into the segmentation pipeline. By leveraging a transformer-based mechanism, our method adaptively identifies the intricate details of 3D shapes, calculating varying degrees of attention to each face. Consequently, the mesh model is downsampled by eliminating faces with lower attention, thereby optimizing the segmentation process. Our approach outperforms most state-of-the-art methods on multiple public datasets, making it a promising avenue for future research.**

***Index Terms*—3D shape segmentation, Deep neural network, Nonuniform downsampling, Transformer**

I. Introduction

3D shape segmentation, which involves partitioning 3D shapes into semantic parts, is a crucial aspect of 3D shape understanding and has significant implications for computer vision, computer graphics, robotics, and mixed realities. A wide range of tasks, such as 3D mesh reconstruction, 3D shape deformation, 3D shape classification, and 3D shape retrieval, require accurate and efficient 3D shape segmentation algorithms to achieve satisfactory performance. As a result, 3D shape segmentation has drawn considerable attention in recent years. However, the complexity of 3D shapes has made it an ongoing challenge, despite the significant research efforts devoted to this area.

The majority of existing 3D shape segmentation methods rely heavily on geometric similarity between faces to classify them into meaningful segments. Therefore, extracting robust and effective geometric features for each face is essential in improving the performance of 3D shape segmentation methods. Previous approaches have primarily utilized established 3D shape feature descriptors, such as Shape Diameter Functions [1] (SDF), Average Geodesic Distance [2] (AGD), and Gaussian Curvature [3] (GC), to describe the geometric features of each face. However, a single feature descriptor can only capture the features of faces in one aspect, which significantly hampers the performance of 3D segmentation algorithms. As a result, later methods [4], [5], [6] have sought to combine multiple feature descriptors to achieve better performance than using a single one.

The rapid development of machine learning techniques has led to an increasing number of 3D shape segmentation methods that employ machine learning approaches, particularly deep learning, to obtain more reliable geometric features. These methods can be broadly classified into two main categories. The first category involves the use of a deep neural network to map existing low-level geometric features to high-level ones, as exemplified in [5], [6]. Typically, this type of method relies on a large volume of high-quality labeled training 3D shapes to achieve satisfactory segmentation results. However, manually labeling each face of 3D shapes is widely regarded as an arduous and expensive task. The second category of methods [7], [8], [9] involves projecting 3D shapes into multiple 2D views and transforming the task of 3D shape segmentation into 2D image segmentation. Capitalizing on the transfer of prior knowledge learned from existing 2D image datasets, this category of methods exhibits superior performance compared to other approaches. However, it is susceptible to occlusions that arise during projections, which limits its performance improvement.

In this paper, we propose a novel 3D shape segmentation method that incorporates attentive nonuniform downsampling. Prior approaches that utilize downsampling typically employ a uniform approach, resulting in significant information loss across faces, particularly those in regions of high curvature or intricate details. Our method, however, leverages an attentive module to automatically calculate the attention value for each face in the mesh model, thereby preserving more faces in detail areas during the downsampling phase. As a result, our approach exhibits a significant advantage over existing segmentation methods. As shown in Figure 1, our method utilizing attentive nonuniform downsampling can perform better than uniform downsampling on boundary areas.

Our contributions are two-fold:

- In this paper, we introduce a novel 3D shape segmentation approach that employs attentive nonuniform downsampling. In contrast to traditional uniform downsampling, our newly designed attentive downsampling method learns the attention value for each face, enabling automatic and efficient selection of areas for downsampling.

Zhenyu Shu and Chaoyi Pang are with School of Computer and Data Engineering, NingboTech University, Ningbo 315100, China. They are also with Ningbo Institute, Zhejiang University, Ningbo 315100, China (e-mail: shuzhenyu@nit.zju.edu.cn, chaoyi.pang@qq.com).

Xufei Sun is with College of Computer Science and Technology, Zhejiang University, Hangzhou 310058, China (e-mail: xufeisun_paper@163.com). Corresponding author.

Shiqing Xin is with School of Computer Science and Technology, Shan-Dong University, Jinan 250100, China (e-mail: xinshiqing@sdu.edu.cn).

Manuscript received month day, year; revised month day, year.

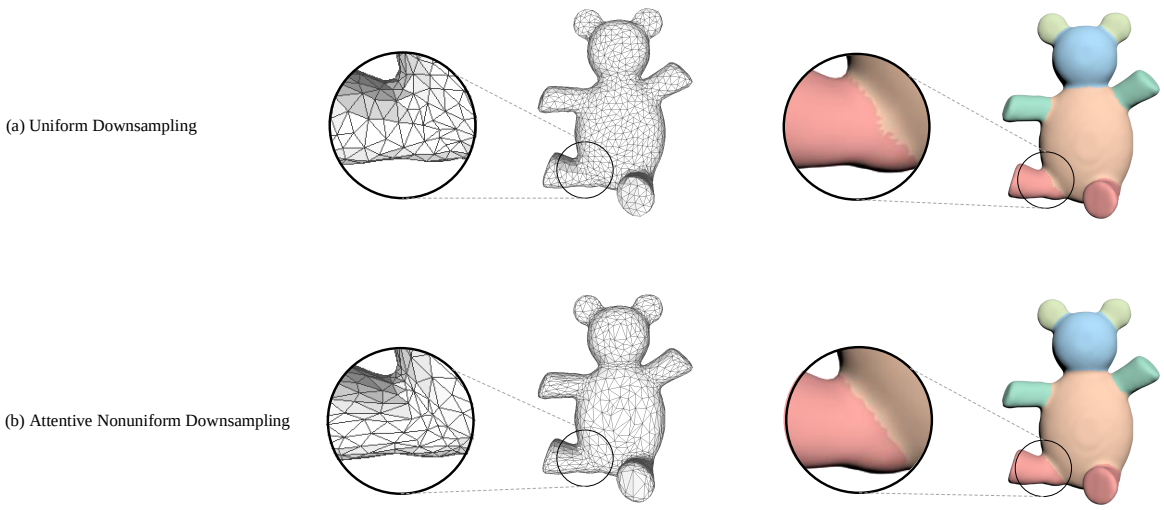

Fig. 1. In most mesh models, the triangular faces are not uniformly placed. In areas with larger curvature, especially in the junctions between different segments, the faces are usually smaller and denser, while on the other hand, faces in flatter areas are usually larger and more sparse. Treating faces with equal attention is not the most efficient way in order to segment the 3D mesh model, and thus nonuniformly sample the model, subtract faces in the flatter areas, which correspondingly add more attention to those faces with greater curvature.

- Extensive experiments on various datasets demonstrate that our method outperforms previous approaches.

The remaining parts of this paper are organized as follows. First, we introduce related work in Section II. Second, we describe the details of our method in Section III. Third, Section IV shows the performance of our method and compares it to state-of-the-art methods on public benchmarks. Fourth, the limitations and future work of our method are explained in Section V. Finally, we conclude our paper in Section VI.

II. Related Work

Shape segmentation, the process of partitioning 3D shapes into meaningful semantic parts, is a critical research area in computer vision, computer graphics, robotics, and mixed realities. 3D shapes can be represented using three main representations: surface meshes, point clouds, or voxels. This section reviews surface mesh-based methods for 3D shape segmentation, while some related approaches for point clouds are also introduced here.

A. Traditional segmentation methods

Early approaches to 3D shape segmentation primarily involved the use of hand-crafted feature descriptors. These feature descriptors were designed to map all faces into a feature space and subsequently apply clustering algorithms to divide them into several classes for segmentation. Naturally, faces with the same label in a 3D shape should exhibit similar geometric features. AGD, which is calculated by averaging the geodesic distance between each vertex and all other vertices, provides global position information of 3D shapes. SDF measures the diameter of the local shape of the face to identify the thin and fat parts of the 3D shape. GC describes the bending degree of each vertex in the 3D shape. Numerous studies have demonstrated that using these feature descriptors and others yields satisfactory results in 3D shape segmentation. To further enhance the performance of segmentation, [10], [11], [12], [13], [14] have proposed combining multiple feature descriptors to extract features of 3D shapes from multiple aspects.

B. Supervised segmentation methods

With the rapid development of 3D shape repositories and machine learning techniques, particularly deep learning, an increasing number of researchers have been investigating supervised learning-based segmentation methods. These methods have been found to be superior to traditional and unsupervised methods due to their ability to learn the mapping relationship from feature vectors to labels through prior knowledge. As a result, many researchers are now focusing on these methods.

The first supervised learning-based method for 3D shape segmentation is introduced by Kalogerakis et al. [4]. They design an objective function with learnable parameters based on the Conditional Random Field (CRF) model, which was optimized by using manually labeled shapes. Similarly, Kaick et al. [14] propose a novel shape segmentation approach that utilizes knowledge by analyzing geometric similarity between matched shapes. Other researchers have used supervised learning methods on multiple geometric feature descriptors to segment shapes. For instance, Xie et al. [15] propose a fast segmentation method on the mesh using Extreme Learning Machines, while Guo et al. [5] employ deep convolutional neural networks to transform multiple geometric feature descriptors into a two-dimensional matrix for 3D shape segmentation. Su et al. [16] present a multi-prototype classifier for 3D point cloud segmentation, with each prototype representing the classifier weights for a specific subclass and incorporating two constraints to update the prototypes and promote diverse learning. Experimental results affirm the effectiveness

of the proposed method, especially in low-label scenarios, and demonstrate the discovery of semantic subclasses without requiring additional annotations. Zhao et al. [17] introduce JS-Net++, a novel approach for 3D point cloud segmentation that integrates instance and semantic segmentation. Their method utilizes a joint module to fuse features from various layers of the backbone network, enabling mutual benefits between the two tasks. Song et al. [18] introduce a hybrid semantic affinity learning method for 3D point cloud segmentation, which captures label dependencies by combining global priors from structural correlations via a graph convolutional network and local affinity to model semantic similarities within and between classes. Their approach enhances the performance of state-of-the-art models across various datasets, including indoor, outdoor, and synthetic environments. Zhang et al. [19] combines the concept of sparse prior, achieved through a differentiable sparse encoding sub-network and a semantic feature extraction sub-network, showing significant improvement in multiple evaluation metrics. SCMS-Net [20] can effectively segment three-dimensional meshes through self-supervised learning without the need for a large amount of annotated data, with high efficiency and accuracy. Wang et al. [21] propose a 3D mesh instance segmentation method that integrates 2D and 3D data, and utilizes the rich information of two-dimensional images and the geometric features of three-dimensional meshes, achieving precise instance segmentation. Laplacian2Mesh [22] applies Spectral Transformation to mesh understanding tasks and leverages the Laplacian spectral theory to manage the irregularities inherent in polygonal meshes. DGNet [23] presents a novel approach to deep neural network (DNN) processing for arbitrary mesh structures. The method, referred to as DGNet, addresses common challenges in mesh processing, such as handling non-manifold geometries and irregular structures that complicate hierarchical feature aggregation. Shi et al. [24] introduce three innovative modules designed to extract diverse temporal information from both local and global contexts, effectively improving feature representation in target frames. Their approach demonstrates superior performance in 3D point cloud semantic segmentation across SemanticKITTI and SemanticPOSS datasets.

In addition to feature-based methods, view-based methods have also been applied to 3D shape segmentation by establishing a connection between 3D shapes and their 2D projection collections. Wang et al. [7] label each projection using the knowledge learned from labeled projections, and then project the labels back onto the mesh for segmentation. Kalogerakis et al. [9] use an image-based Fully Convolutional Network to label the projections, resulting in excellent segmentation results. Le et al. [25] propose a method that treats multiple 2D projections of the 3D shape as a sequence and uses Recurrent Neural Networks (RNNs) for segmentation. Similarly, MeshWalker [26] also uses RNNs for 3D shape segmentation, but their sequences were obtained by random walking on the mesh surface. Chang et al. [27] introduce a novel multi-phase fusion network for 3D point clouds, integrating weakly supervised loss, attention-based feature fusion, and self-confidence-based late fusion at the pixel level. Their approach achieves competitive results on nuScenes and SemanticKITTI benchmarks, showcasing its superior performance. Rong et al. [28] introduce a novel framework for 3D semantic segmentation of aerial photogrammetry models, leveraging orthographic projection to enhance efficiency without compromising precision. Their approach is versatile and applicable across various types of models in the field.

C. Transformer and self-attention

Bahdanau et al. [29] propose a class attention mechanism for simultaneous translation and alignment in machine translation, which is widely recognized as the first work to apply attention to natural language processing. Later, Lin et al. [30] propose and apply the self-attention mechanism to the visualization and interpretation of statement embeddings. Building on the self-attention mechanism, Vaswani et al. [31] propose the Transformer model for machine translation. The Transformer model is solely based on self-attention, without using recurrent neural networks or convolutional operators. Since its inception, Transformer has achieved excellent results in many natural language processing problems. Compared to previous methods, it trains faster and establishes more effective long-distance dependency relationships. The Transformer model has also inspired the development of many pre-training models, including BERT proposed by Devlin et al. [32]. BERT uses a bidirectional Transformer to pre-train deep bidirectional representations by jointly adjusting the left and right contexts at all levels. BERT achieves first place in all 11 natural language processing tasks and has received tremendous feedback in the field. Encouraged by Transformer and BERT, many excellent methods have been proposed to further extend the Transformer framework, such as Transformer XL [33] and BioBERT [34].

Given the remarkable success of self-attention mechanisms in natural language processing, researchers have begun to explore their potential application in two-dimensional computer vision. Before introducing of self-attention mechanisms into two-dimensional vision, convolutional neural networks were one of the major frameworks dominating the field. Researchers initially attempted to incorporate self-attention layers into convolutional neural networks to capture long-distance relationships, such as in GCNet [35] and Jie et al. [36]. Other researchers attempt to abandon the mainstream convolutional neural network architecture and instead use purely self-attention mechanisms, such as Parmar et al. [37], Hu et al. [38], and Zhao et al. [39]. The Vision Transformer (ViT) proposed by Dosovitskiy et al. [40] further extends this pure self-attention architecture to large-scale pre-training, achieving optimal results in many two-dimensional tasks. ViT's excellent results suggest that, just as pure self-attention methods outperform traditional recurrent neural networks in natural language processing, pure self-attention mechanisms can also outperform traditional convolutional neural network architectures in two-dimensional vision. Currently, ViT has become a landmark method in two-dimensional vision algorithms.

With the remarkable success of self-attention mechanisms in both natural language processing and two-dimensional

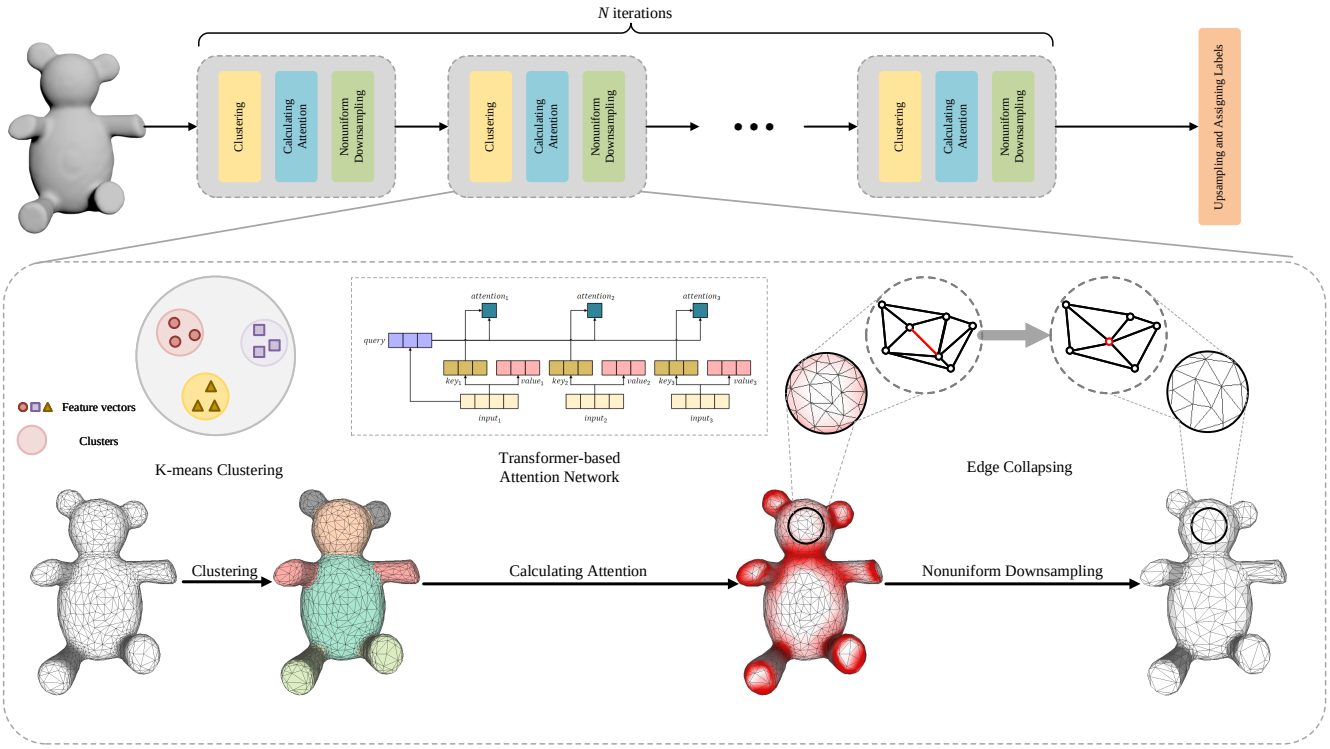

Fig. 2. Our method takes the original mesh model as input, followed by several units, each containing three different modules. In each unit, the input model is the output of the former unit, and firstly, K-means clustering is conducted on the input model. Based on the clusters, attention is assigned to each face via a transformer module, and lastly, the model is nonuniformly downsampled according to the attention. We select the faces to be deleted based on the attention value, and then perform an edge collapsing operation to the shared edge of two faces with the smallest attention (colored in red) and modify the adjacent faces based on the newly generated vertex (colored in red).

computer vision, researchers have naturally sought to extend them to three-dimensional shapes. However, unlike statements in natural language processing that have a natural order and pixels in two-dimensional images that have an up, down, left, and right order, point clouds and mesh models in three-dimensional shapes lack orderliness. Therefore, solving the disorder problem has become crucial in applying self-attention mechanisms to three-dimensional shapes. In 2021, Guo et al. [41] propose the Point Cloud Transformer (PCT), which is the first to apply self-attention mechanisms to 3D point clouds. The PCT successfully solves the problem of point cloud disorder by using input embedding based on 3D coordinates, providing great inspiration for the field of 3D point clouds. Subsequently, Zhao et al. [42] propose the Point Transformer, which uses coordinate differences between vertices to design learnable position encoding. Both methods utilize the order invariance of point clouds and employ farthest point sampling and nearest neighbor search.

III. Our Method

In this section, we will introduce the details of our method. As shown in Figure 2, our method consists of three modules, including face clustering, attention, and nonuniform downsampling, with an overall loop of four times. After these steps, our output is a mesh model downsampled based on our trained attention, and then we conduct segmentation on this mesh model. Finally, we perform an upsampling operation with reference to previous downsampling paths and get the final predicted labels for the original model.

In the following, we introduce the details of each module in our segmentation pipeline. Section III-A and Section III-B describe the details of face clustering and attentive nonuniform down-sampling modules, respectively. Section III-C explains the upsampling unit, and Section III-D concludes the overall process of our algorithm.

A. Clustering module

Clustering is a classical machine learning problem that involves segmenting a series of unlabeled data into different classes or clusters according to specific criteria. In our clustering module, we focus on dividing faces in mesh models into different clusters.

Given a 3D mesh model $M = \{V, E, F\}$, where V represents vertices, E represents edges, and F represents faces, we firstly calculate the feature vector x_i for each face f_i. By using the feature vector x_i of each face, we can cluster the faces using a simple K-means clustering algorithm to obtain a label for each face. Faces with the same label belong to the same cluster.

During the downsampling process, excessive consumption of global feature calculations can impede efficiency. Therefore, we use local information features as feature descriptors to reduce computational complexity. Specifically, we select the length of three edges of each face, namely a, b, c, to encode

the geometric information of the triangular face. In order to eliminate the impact caused by different orders of edges, we use the edge-encoded vector $(a+b+c, ab+bc+ca, abc)$ which is circulant symmetric and at the same time able to solve the value of a, b and c. Furthermore, we add three additional feature descriptors, including GC, SDF, and AGD to form the feature vector.

B. Attentive downsampling module

The attentive downsampling module is designed to downsample 3D models while retaining essential information, making it a critical component of many 3D shape processing pipelines. The module comprises two distinct parts: the attention module and the downsampling module. The attention module is responsible for identifying the most informative faces in a given mesh model, while the downsampling module reduces the number of faces in the model while retaining the relevant information. The attention module uses an attention mechanism to automatically assign weights to each face based on its importance to the overall shape of the mesh model. The weights of the faces are then used to guide the downsampling process, ensuring that the most informative faces are retained.

The attention module assigns importance weights to individual faces in a mesh model. The primary task of the attention module is to calculate attention values for each face, which is accomplished using a transformer-based training approach. Our methodology involves generating a sequential arrangement of patches based on the clustering order of the faces, which serves as input to the attention module. This approach enables us to customize the attention values across the faces, ensuring they are processed and analyzed appropriately. Specifically, the transformer-based training approach uses self-attention mechanisms to calculate attention values for each face. During training, the model learns to weigh the importance of each face based on its contribution to the overall shape of the mesh model. The attention module can effectively capture important features and retain essential information while downsampling the mesh model. The transformer-based attention module can be represented by the following equations.

$$(Q, K, V) = X \cdot (W_Q, W_K, W_V), \tag{1}$$

$$\tilde{A} = Q \cdot K^T, \tag{2}$$

$$\bar{A} = \frac{\tilde{A}}{\sqrt{d_F}}, \tag{3}$$

$$A = \text{Softmax}(\bar{A}), \tag{4}$$

where W_Q, W_K, W_V represents learnable parameter matrix, X represents the input feature matrix, d_F is the dimension of input features.

To achieve nonuniform sampling in the attentive downsampling module, we use the attention values generated in the attention module to sort each face in the mesh model from small to large. Specifically, we select the face with the smallest attention value, and then choose the adjacent face with the smallest attention value from the three faces that share an edge with the selected face. We then perform attentive edge collapsing on these two faces and their adjacent edge, which eliminates the two selected faces. This process is repeated iteratively until the desired number of faces is reached. Figure 3 illustrates this process, where att_i represents the attention score of each face, and assuming $att_i < att_j$ if $i < j$. From all the faces, we select the one with the least attention, which is face abc. We then select the surrounding face bcd with the smallest attention value, and collapse the shared edge bc into one newly generated point e, resulting in the elimination of the two selected faces. This process is repeated iteratively until the desired number of faces is reached. Using attention values to guide the downsampling process in this manner allows for nonuniform sampling, where the most informative faces are retained while the number of faces is reduced.

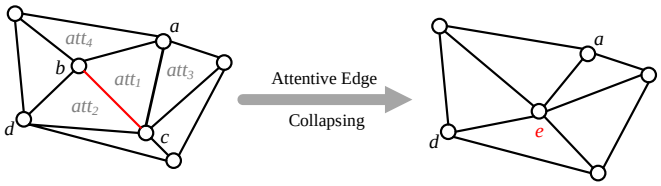

Fig. 3. The sketch of the attentive edge collapsing process. As shown in the sketch, the attention value of each face is noted as att_i, and we assume $att_i < att_j$ if $i < j$. Two faces are shown in the image, and the edge cd color in red is the shared edge, which is to be collapsed. A new vertex e is formed in the midpoint of cd, and vertices a and b are connected to the new vertex.

The attentive edge collapsing operation is iteratively applied until the number of downsampled faces reaches half of the total faces in the input model. The resulting downsampled 3D model then serves as the input for the subsequent processing unit. The progression of our downsampling operation is visually represented in Figure 4, which shows the gradual reduction in the number of faces while preserving critical features based on their attention values. The upper row of the figure shows the uniform downsampling procedure, while the lower row shows our attentive nonuniform downsampling procedure. In the figure, the shade of red represents the attention values of each face, where a darker shade of red indicates a higher attention value, and a lighter shade of red indicates a lower attention value. The figure shows that during the uniform downsampling process, the attention distribution on the model is relatively broad, and not all are concentrated in the detailed parts or boundary parts. In the nonuniform downsampling process, the high attention values are mainly distributed at the detailed parts and boundary parts, which allows our method to preserve more faces in these areas during the downsampling process. At the same time, it can be observed that during the nonuniform downsampling process, the attention distribution becomes increasingly concentrated as the downsampling progresses, which also proves that our nonuniform downsampling process effectively preserves the necessary information.

The above three modules, the clustering module, attention

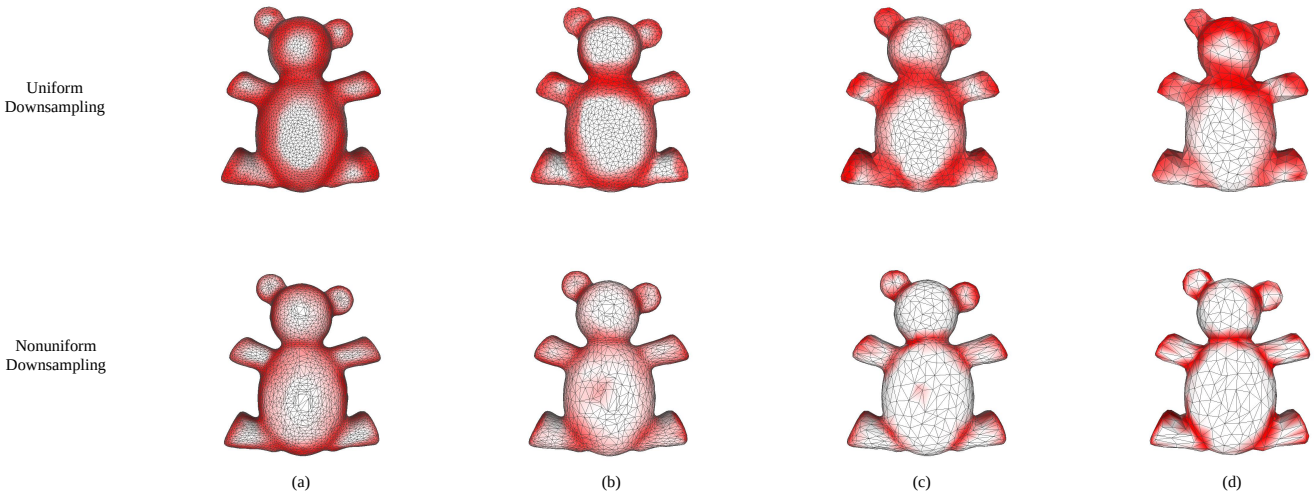

Fig. 4. Comparison between uniform and nonuniform downsampling processes. The upper ones are those performing uniform downsampling, and the lower ones are performing nonuniform downsampling. The original mesh model contains approximately 24000 faces in total, and models (a), (b), (c), and (d) show the result of the downsampled model to the number of faces of around 12000, 6000, 3000, and 1500, respectively. The heat map colors on each model represent the high and low attention values, with darker red indicating larger attention values. It can be seen that in the attentive nonuniform downsampling process, higher attention values are more focused on parts which are more detailed or on the boundary, whereas in uniform downsampling, attention distribution is more dispersed.

module, and downsampling module, are sequentially combined as a unit, and the final downsampled mesh model is obtained by cycling N times as a whole. In this paper, we experimentally set $N = 4$. An ablation study about different choices of the value of N can be found in Section IV-C.

C. Upsampling unit

After performing four cycles of attentive nonuniform downsampling, we obtain a downsampled 3D model. Subsequently, a final clustering operation is executed on this downsampled model to assign labels to each face. To reintegrate these labels into the original model, we perform an inverse operation by retracing the steps of the previous downsampling path. Specifically, we perform attentive edge expansion, which involves the addition of new faces to the downsampled model by inverting the attentive edge collapsing operation used during downsampling. This process is performed iteratively until we restore the original model. Finally, we assign the labels obtained from the final clustering operation to each face in the original model, providing a complete and labeled 3D model.

The attentive nonuniform downsampling technique has unique characteristics that distinguish it from other downsampling techniques and make it an effective approach for preserving the local structure of 3D meshes during the downsampling and upsampling phases of mesh processing. Specifically, the attentive nonuniform downsampling technique ensures that when faces are collapsed during downsampling, they are predominantly located within each cluster rather than at the intersections between clusters. As a result, during the upsampling phase, the labels of newly generated faces can be directly assigned based on the labels of their surrounding faces, which are likely to be within the same cluster. However, in cases where there are discrepancies among the labels of surrounding faces, a voting mechanism is employed to determine the labels for the newly generated faces. This approach capitalizes on the inherent spatial coherence within clusters, leveraging the localized nature of nonuniform sampling to accurately infer the labels of faces during the upsampling process.

D. Algorithm

Our algorithm is trained and tested on each category of 3D shapes, which can be summarized as Algorithm 1.

IV. Experiments

This section presents the experimental results of the proposed method and provides a comparative analysis with current state-of-the-art approaches. Additionally, a series of ablation experiments are conducted to validate the efficacy and rationality of the proposed approach. The experimental results demonstrate the superior performance of the proposed method, and the ablation experiments provide insights into the key components of the approach that contribute to its success.

A. Experimental Setting

Dataset. We employ the Princeton Segmentation Benchmark [43] (PSB), the COSEG benchmark [44], and the Human Body Dataset proposed by [45] in the experiments to evaluate our algorithm. PSB and COSEG are the two most popular datasets for benchmarking 3D shape segmentation algorithms. The PSB dataset contains 19 categories, with 20 models for each category. We remove the three categories of bust, bearing, and mech because the models in these categories lack consistent semantic labels. The small dataset of the COSEG contains shapes for eight classes, and the large dataset consists of three classes. The Human Body Dataset is a newly constructed and

Algorithm 1 3D Shape Segmentation via Attentive Nonuniform Downsampling

Inputs: Training 3D shapes and human-assigned labels for all faces.

Outputs: Predicted label for each face on test 3D shapes.

1: For each input 3D shape, calculate the feature vectors of each face, including the triangle's inner angle, edge-length ratio, and feature descriptors GC, SDF, and AGD;
2: **repeat**
3: Conduct K-means clustering operation to all faces using the feature vectors;
4: Generate face sequences based on clustering results and calculate the attention value of each face through the transformer module;
5: Starting from the face with the smallest attention, select one face at a time and conduct an attentive edge collapsing between the face selected and one of the adjacent faces with the smallest attention. Repeat until the total number of faces is half of the input 3D shape;
6: **until** Iteration over.
7: For the output 3D shape, calculate the feature vectors for each face and cluster them. Perform the inverse operation according to the path of each nonuniform downsampling, assign labels to the upsampling generated faces based on the clustering labels of the surrounding faces, until the 3D shape is restored to the original 3D model;
8: Smooth the segmentation results using graph-cuts.
9: **return** All labels of clustered faces.

TABLE I
THE ACCURACY OF SEGMENTATION RESULTS FOR EACH CATEGORY OF 3D SHAPES IN PSB DATASET COMPARED WITH THREE OTHER METHODS, INCLUDING SHAPEPFCN [9], MESHCNN [46], AND MESHWALKER [26], ON THE PSB DATASET.

Category	ShapePFCN	MeshCNN	MeshWalker	Ours
Cup	93.70%	95.86%	**99.54%**	99.39%
Table	99.30%	96.78%	99.33%	**99.47%**
Teddy	96.50%	84.29%	95.57%	**97.80%**
Bird	86.30%	68.09%	**92.76%**	92.09%
Hand	**88.70%**	68.83%	83.31%	88.61%
Fish	95.90%	89.05%	94.58%	**96.18%**
Human	93.80%	74.76%	87.02%	**94.31%**
Glasses	96.30%	93.94%	96.11%	**96.94%**
Airplane	92.50%	84.36%	96.20%	**96.93%**
Ant	**98.90%**	91.83%	97.36%	98.66%
Chair	98.10%	84.75%	97.61%	**98.72%**
Octopus	98.10%	**98.21%**	97.86%	98.05%
Plier	95.70%	83.69%	92.24%	**96.51%**
Armadillo	93.30%	50.24%	89.12%	**93.85%**
Vase	85.70%	68.94%	84.56%	**87.03%**
FourLeg	89.50%	68.73%	80.93%	**90.10%**
Average	93.89%	81.40%	92.76%	**95.29%**

recently popular dataset formed by 381 training models and 18 testing models. The division of the training and validation sets for PSB is referenced from [5]. We take 12 models as the training sets for each category and the rest as the validation sets.

Experiment details. We implement our algorithm in Python and Matlab. In our network, the initial weights are set to variables subject to a Gaussian distribution with a variance of 0.001 and a mean of zero. The optimizer is Adam, with a learning rate of 0.001. Our algorithm runs on a single NVIDIA GeForce RTX 3090 GPU. With the consumption of shape preprocessing, for each model with 20K-30K faces, our algorithm needs 10 minutes for training and 30 seconds for evaluation.

TABLE II
THE ACCURACY OF SEGMENTATION FOR EACH CATEGORY OF 3D SHAPES IN THE SMALL COSEG DATASET COMPARED WITH THREE OTHER METHODS, INCLUDING SHAPEBOOST [4], MESHCNN [46], AND SHAPEPFCN [9].

Category	ShapeBoost	MeshCNN	ShapePFCN	Ours
Candelabra	85.50%	83.52%	**95.40%**	93.40%
Chairs	94.80%	92.87%	96.10%	**96.64%**
Fourleg	92.30%	86.19%	90.40%	**93.37%**
Goblets	97.00%	92.62%	97.20%	**97.92%**
Guitars	97.70%	91.34%	**98.00%**	97.85%
Irons	87.20%	81.26%	88.00%	**88.69%**
Lamps	76.30%	83.64%	**93.00%**	92.41%
Vases	86.40%	77.43%	84.80%	**88.13%**
Average	89.65%	86.11%	92.86%	**93.55%**

B. Results and Comparison

In this study, we conduct experiments to evaluate the performance of our proposed method. To assess the effectiveness of our method, we use the widely adopted metric in the field. Similar to Guo et al. [5], we use the following segmentation accuracy metric to evaluate the performance of our approach:

$$Accuracy = \sum_{i \in T} t_i \mathbf{u}\left(l_i\right) / \sum_{i \in T} t_i, \tag{5}$$

where T is the face set of the testing 3D shapes, t_i is the area of the face i, and l_i is the predicted label of face i. $\mathbf{u}\left(l_i\right)$ is equal to 1 if the prediction is correct, otherwise, it is 0.

Table I presents the accuracy of our method on the PSB dataset. Table II shows the accuracy of our method on the COSEG dataset, and the accuracy on the Human Body dataset is presented in Table III. We obtain an average accuracy of 95.29% on the PSB dataset, 93.55% on the small COSEG datasets respectively, and 93.12% on the Human Body dataset. Figure 5, Figure 6, and Figure 7 show some samples of the segmentation results of our method on the PSB, COSEG, and Human Body datasets, respectively. Figure 8 shows the comparison between the segmentation results of our method and the ground truth. It can be seen that our segmentation results are very close to the ground truth.

Furthermore, we visually compared our method with MeshCNN [46] and Guo et al. [5], as shown in Figure 9. In the figure, we visually compared our method with MeshCNN and Guo et al.'s method on a Fourleg model with more detailed information and a Hand model with many small boundaries. It can be seen that our method outperforms the other two methods in both segmenting detail parts of models and processing boundary parts of models, proving the superior performance of our method.

TABLE III
THE ACCURACY OF SEGMENTATION ON THE HUMAN BODY DATASET COMPARED WITH SIX OTHER METHODS, INCLUDING MARON ET AL. [45], DIFFUSIONNET [47], FIELD CONVOLUTIONS [48], HODGENET [49], MDGCNN [50] AND PFCNN [51].

Method	Maron et al.	DiffusionNet	Field Convolutions	HodgeNet	MDGCNN	PFCNN	Ours
Accuracy	88%	90.80%	92.90%	85.03%	89.47%	91.79%	**93.12%**

Fig. 5. The samples of segmentation results on the PSB dataset.

Fig. 6. The samples of segmentation results on the COSEG dataset.

Our experimental results show that our proposed method achieves accuracy levels that outperform state-of-the-art algorithms in most categories, highlighting its superior performance compared to other methods.

C. Ablation studies

The key idea of our algorithm is to perform attentive nonuniform downsampling on three-dimensional shapes to obtain more accurate segmentation results. Thus, our ablation experiments include three aspects.

TABLE IV
THE COMPARISON BETWEEN ACCURACY OF SEGMENTATION USING NONUNIFORM DOWNSAMPLING ,UNIFORM DOWNSAMPLING AND WITHOUT DOWNSAMPLING.

Category	Nonuniform	Uniform	Without downsampling
Human	**94.31%**	91.74%	89.20%
Teddy	**97.80%**	95.57%	92.65%
Airplane	**96.93%**	95.79%	92.48%

Firstly, to verify the effectiveness of nonuniform downsampling, we conduct ablation study among nonuniform downsampling, uniform downsampling and without downsampling. The nonuniform downsampling module was replaced by a uniform downsampling module that does not rely on attention for testing to generate results of uniform downsampling. The segmentation results of without downsampling is formed by deleting the attentive module as well as the downsampling module, and segment only using the features. The samples of results compared with ground truth are shown in Figure 10, and the quantitative results are shown in Table IV. These results show the advantage of our proposed nonuniform downsampling, especially in regions of junctions of different segments. Compared to the method without downsampling, nonuniform downsampling helps achieve better segmentation results because, during the nonuniform downsampling process, our method folds patches with low attention values and applies more attention to more important patches, such as small parts and intersections. Therefore, our method can improve the segmentation results in these parts, leading to an overall

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2024.3432095

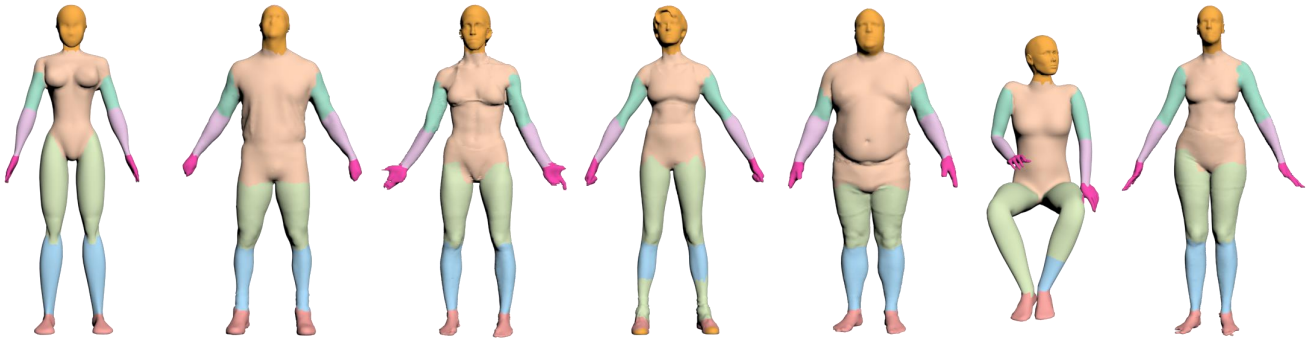

Fig. 7. The samples of segmentation results on the Human Body dataset.

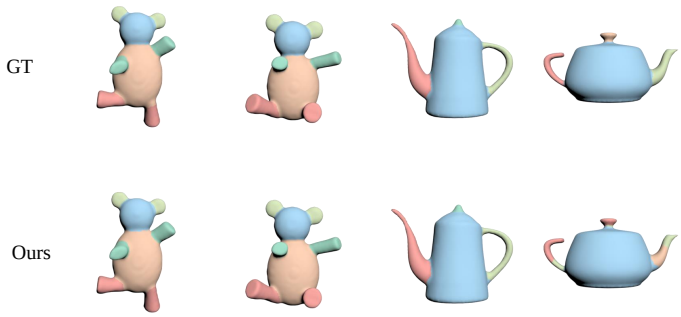

Fig. 8. The comparison on the PSB dataset between our segmentation result ("Ours" in the image) and the ground truth ("GT" in the image).

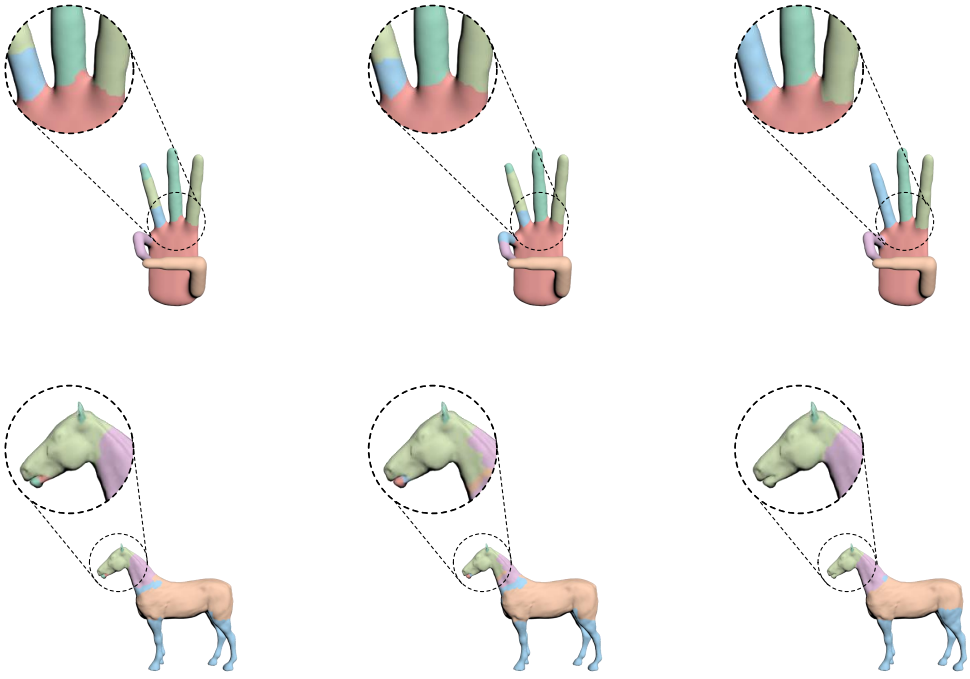

Fig. 9. The comparison of segmentation results among our method, MeshCNN [46], and Guo et al. [5]. Our method outperforms the other two methods in detailed areas and boundary segmentation.

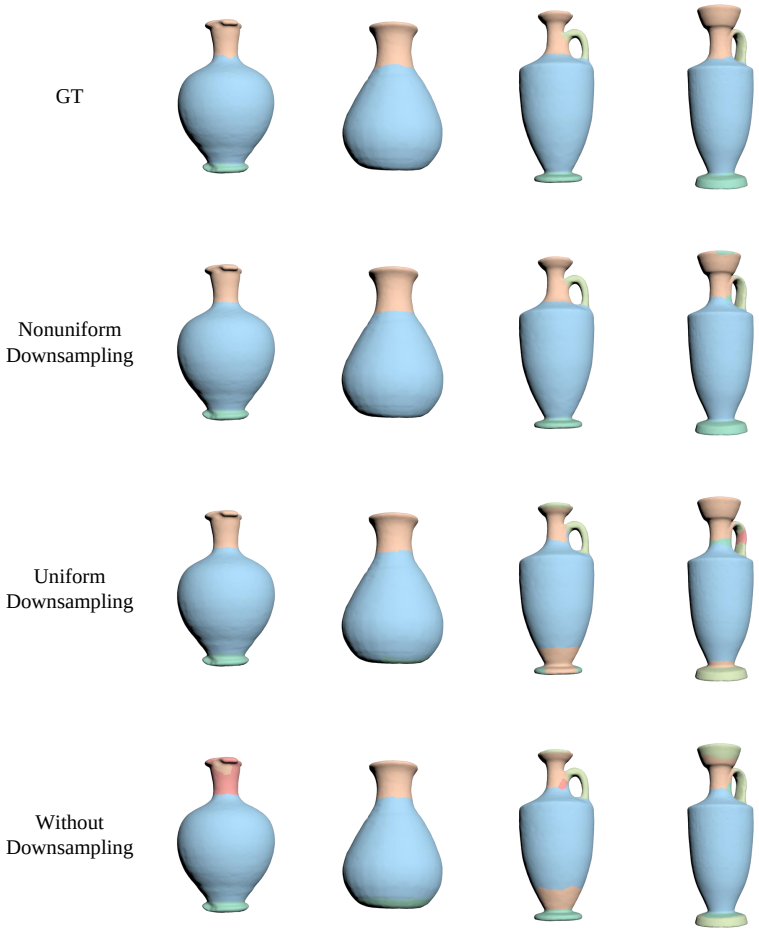

Fig. 10. The comparison of segmentation results among ground truth ("GT" in the image), uniform downsampling, nonuniform downsampling, and without downsampling.

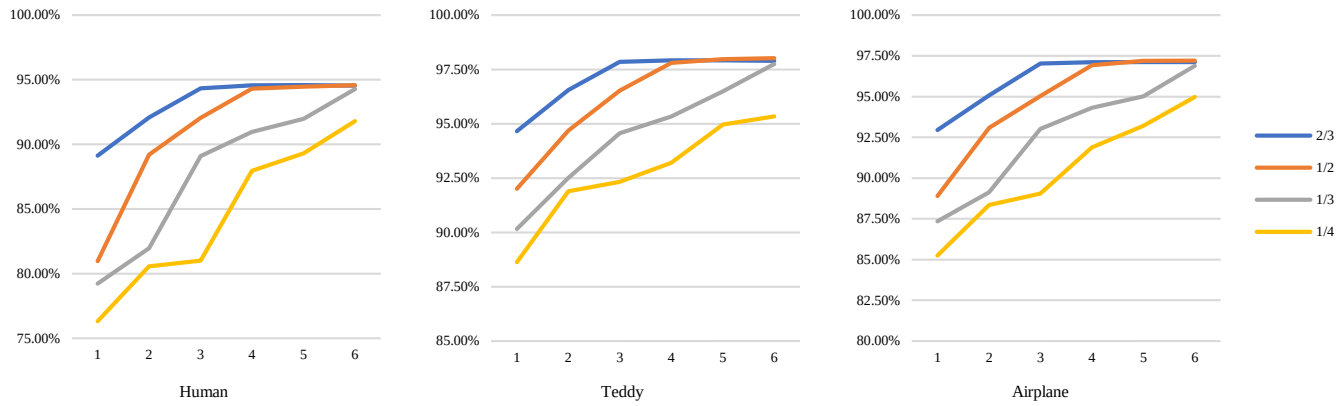

Fig. 11. The comparison of segmentation accuracy with different numbers of iterations and different downsampling ratios in each iteration. The horizontal axis represents the number of cycles, and the vertical axis represents the accuracy. The color of lines in the figure shows the ratios of downsampling in each iteration.

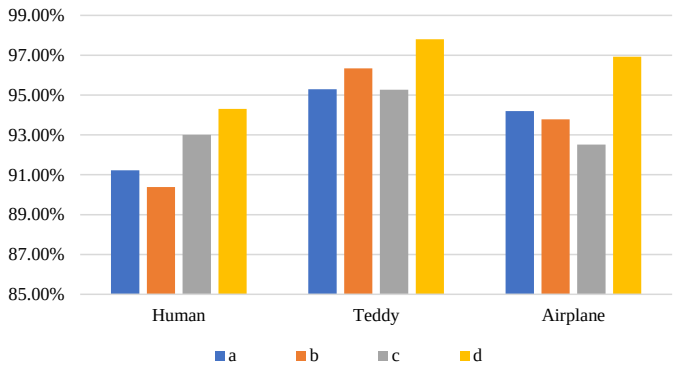

Fig. 12. The comparison of segmentation results using different combinations of feature descriptors. (a) represents using edge-encoded vector only. (b) represents edge-encoded vector, and GC. (c) represents edge-encoded vector, GC, and SDF. (d) represents edge-encoded vector, GC, SDF, and AGD.

improvement in segmentation accuracy.

Secondly, in order to verify the impact of the downsampling ratios as well as the number of cycles on the segmentation results, we used different ratios and numbers of cycles (N) for training and testing, and obtained the results shown in the figure based on the segmentation accuracy. From the change in segmentation accuracy, as shown in Figure 11, it can be seen that for the same downsampling ratio, accuracy generally increases gradually with the number of iterations. However, when the number of iterations reaches a certain level, it becomes slower or even shows a downward trend. This is because for some details, downsampling to a small number of faces can lead to the loss of necessary detailed information, which to some extent affects the final segmentation effect. For different downsampling ratios, it can be seen from the ablation experiment that the larger the downsampling ratio, the better the effect after a single cycle. However, as the number of cycles increases, the total number of faces gradually decreases, and the segmentation accuracy shows different changes. We think the reason is that if the downsampling ratio is too large, the number of faces decreases sharply, and information is lost during the iteration process. If the downsampling ratio is too small, fewer faces are eliminated in each iteration, and the convergence speed slows down. Therefore, we choose a downsampling ratio of 0.5 and a number of cycles of 4 as the hyperparameters for the experiments in our paper.

Thirdly, to verify the optimality of clustering feature selection, several sets of selected features were recombined and tested. We design four different combinations of feature descriptors totally. The first one is the edge-encoded vector. The second one combines edge-encoded vector, and GC. The third combines edge-encoded vector, GC, and SDF. The last combines edge-encoded vector, GC, SDF, and AGD. Figure 12 shows evidence that using edge-encoded vector, as well as GC, SDF, and AGD achieve the best performance. Thus, we believe that using the combination of all features mentioned above help express more express more aspects of geometric features and lead to an improvement of segmentation results. We think the possible explanation for the different effects of using various feature descriptor combinations is as follows: each feature descriptor focuses on different aspects of information, and different feature descriptors might perform differently on different categories of 3D shapes or even cause conflicts. More specifically, the edge-encoded vector, which is obtained by combining the lengths of three edges of a triangle, can only represent features of the single triangular face, not providing enough effective information for segmentation, therefore using only the edge-encoded feature descriptor results in poorer segmentation performance. GC mainly measures curvature properties and can provide useful information in areas with sharp edges or corners. Therefore, the segmentation results from GC as the feature descriptor may not be satisfactory for 3D shapes in the category of Human and Airplane, which are mostly smooth surfaces. In addition, there may be situations in the Human category where curvatures are similar in the parts of the arm and leg, which might cause mis-segmentation. SDF captures the local thickness of an object near a point on the mesh surface by measuring the distance from the point on its relative surface. However, SDF may perform poorly for details such as the junction of the fuselage and wings in the Airplane category and the connection between the torso and limbs in the Teddy category, leading to poor segmentation results. The above feature descriptors mainly focus on the local information of 3D shapes, while AGD can provide global shape information of the entire model by calculating the average geodesic distance from each point on a shape's surface to all other points. In the case where the above feature descriptors depict local information, incorporating the global information encoded by AGD can effectively improve the performance of segmentation results. Therefore, one can see that adding AGD to the feature descriptors leads to a significant performance improvement.

V. Limitations and Future Works

Our algorithm currently faces some limitations. Firstly, feature descriptors need to be computed for each face using our algorithm, which requires the 3D shape to be manifold. We plan on addressing this by extending our approach to non-manifold shapes in the future. Secondly, the computational cost of using our proposed attentive nonuniform downsampling network is relatively high. To mitigate this issue, we will explore more efficient network architectures in our future work.

VI. Conclusion

In this paper, we propose a novel algorithm for segmenting 3D shapes by incorporating the concept of attentive nonuniform downsampling. The complexity of 3D shapes is a pervasive issue, particularly in the conjunctions of different parts of shapes where faces tend to be smaller and denser and thus require more attention. Previous methods have treated all faces equally in every mesh model, whereas our method implements nonuniform downsampling and an attention module that assigns attention to each face in the model, leading to attentive edge collapsing concerning the attention of each face. The experimental results on PSB, COSEG, and Human Body benchmarks demonstrate that our approach surpasses previous methods.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (62172356, 61872321), Zhejiang Provincial Natural Science Foundation of China (LY22F020026), the Ningbo Major Special Projects of the "Science and Technology Innovation 2025" (2020Z005, 2020Z007, 2021Z012).

REFERENCES

[1] L. Shapira, A. Shamir, and D. Cohen-Or, "Consistent mesh partitioning and skeletonisation using the shape diameter function," *The Visual Computer*, vol. 24, no. 4, pp. 249–259, 2008.

[2] L. Shapira, S. Shalom, A. Shamir, D. Cohen-Or, and H. Zhang, "Contextual part analogies in 3D objects," *International Journal of Computer Vision*, vol. 89, no. 2, pp. 309–326, 2010.

[3] R. Gal and D. Cohen-Or, "Salient geometric features for partial shape matching and similarity," *ACM Transactions on Graphics*, vol. 25, no. 1, pp. 130–150, 2006.

[4] E. Kalogerakis, A. Hertzmann, and K. Singh, "Learning 3D mesh segmentation and labeling," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 1–12, 2010.

[5] K. Guo, D. Zou, and X. Chen, "3D mesh labeling via deep convolutional neural networks," *ACM Transactions on Graphics*, vol. 35, no. 1, pp. 1–12, 2015.

[6] Z. Shu, C. Qi, S. Xin, C. Hu, L. Wang, Y. Zhang, and L. Liu, "Unsupervised 3D shape segmentation and co-segmentation via deep learning," *Computer-Aided Geometric Design*, vol. 43, pp. 39–52, 2016.

[7] Y. Wang, M. Gong, T. Wang, D. Cohen-Or, H. Zhang, and B. Chen, "Projective analysis for 3D shape segmentation," *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 1–12, 2013.

[8] Z. Xie, K. Xu, W. Shan, L. Liu, Y. Xiong, and H. Huang, "Projective feature learning for 3D shapes with multi-view depth images," *Computer Graphics Forum*, vol. 34, no. 7, pp. 1–11, 2015.

[9] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri, "3D shape segmentation with projective convolutional networks," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2017, pp. 3779–3788.

[10] Q. Huang, V. Koltun, and L. Guibas, "Joint shape segmentation with linear programming," *ACM Transactions on Graphics*, vol. 30, no. 6, p. 1–12, 2011.

[11] O. Sidi, O. van Kaick, Y. Kleiman, H. Zhang, and D. Cohen-Or, "Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering," *ACM Transactions on Graphics*, vol. 30, no. 6, pp. 1–10, 2011.

[12] R. Hu, L. Fan, and L. Liu, "Co-segmentation of 3D shapes via subspace clustering," *Computer Graphics Forum*, vol. 31, no. 5, pp. 1703–1713, 2012.

[13] V. G. Kim, W. Li, N. J. Mitra, S. Chaudhuri, S. DiVerdi, and T. Funkhouser, "Learning part-based templates from large collections of 3D shapes," *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 1–12, 2013.

[14] O. V. Kaick, N. Fish, Y. Kleiman, S. Asafi, and D. Cohen-OR, "Shape segmentation by approximate convexity analysis," *ACM Transactions on Graphics*, vol. 34, no. 1, pp. 1–11, 2014.

[15] Z. Xie, K. Xu, L. Liu, and Y. Xiong, "3D shape segmentation and labeling via extreme learning machine," *Computer Graphics Forum*, vol. 33, no. 5, pp. 85–95, 2014.

[16] Y. Su, X. Xu, and K. Jia, "Weakly supervised 3D point cloud segmentation via multi-prototype learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 12, pp. 7723–7736, 2023.

[17] L. Zhao and W. Tao, "JSNet++: Dynamic filters and pointwise correlation for 3D point cloud instance and semantic segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 4, pp. 1854–1867, 2023.

[18] Z. Song, L. Zhao, and J. Zhou, "Learning hybrid semantic affinity for point cloud segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4599–4612, 2022.

[19] G. Zhang and R. Zhang, "MeshNet-SP: A semantic urban 3D mesh segmentation network with sparse prior," *Remote Sensing*, vol. 15, no. 22, 2023.

[20] X. Jiao, Y. Chen, and X. Yang, "SCMS-Net: Self-supervised clustering-based 3D meshes segmentation network," *Computer-Aided Design*, vol. 160, p. 103512, 2023.

[21] W. X. Wang, G. X. Zhong, J. J. Huang, X. M. Li, and L. F. Xie, "Instance segmentation of 3D mesh model by integrating 2D and 3D data," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVIII-1/W2-2023, pp. 1677–1684, 2023.

[22] Q. Dong, Z. Wang, M. Li, J. Gao, S. Chen, Z. Shu, S. Xin, C. Tu, and W. Wang, "Laplacian2mesh: Laplacian-based mesh understanding," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[23] X.-L. Li, Z.-N. Liu, T. Chen, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Mesh neural networks based on dual graph pyramids," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–14, 2023.

[24] H. Shi, R. Li, F. Liu, and G. Lin, "Temporal feature matching and propagation for semantic segmentation on 3D point cloud sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 12, pp. 7491–7502, 2023.

[25] T. Le, G. Bui, and Y. Duan, "A multi-view recurrent neural network for 3D mesh segmentation," *Computer & Graphics*, vol. 66, pp. 103–112, 2017.

[26] A. Lahav and A. Tal, "MeshWalker: Deep mesh understanding by random walks," *ACM Transactions on Graphics*, vol. 39, no. 6, pp. 1–13, 2020.

[27] X. Chang, H. Pan, W. Sun, and H. Gao, "A multi-phase camera-lidar fusion network for 3D semantic segmentation with weak supervision," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 3737–3746, 2023.

[28] M. Rong and S. Shen, "3D semantic segmentation of aerial photogrammetry models based on orthographic projection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 12, pp. 7425–7437, 2023.

[29] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, 2015.

[30] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *International Conference on Learning Representations*, 2017.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 6000–6010.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019, pp. 4171–4186.

[33] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 2019, pp. 2978–2988.

[34] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, pp. 1234 – 1240, 2019.

[35] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," *IEEE/CVF International Conference on Computer Vision Workshop*, pp. 1971–1980, 2019.

[36] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation networks," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

[37] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 4052–4061.

[38] H. Hu, Z. Zhang, Z. Xie, and S. Lin, "Local relation networks for image recognition," *IEEE/CVF International Conference on Computer Vision*, pp. 3463–3472, 2019.

[39] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10 073–10 082, 2020.

[40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[41] M.-H. Guo, J. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S. Hu, "PCT: Point cloud transformer," *Computational Visual Media*, vol. 7, pp. 187–199, 2020.

[42] N. Engel, V. Belagiannis, and K. C. J. Dietmayer, "Point transformer," *IEEE Access*, vol. 9, pp. 134 826–134 840, 2020.

[43] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 1–12, 2009.
[44] Y. Wang, S. Asafi, O. van Kaick, H. Zhang, D. Cohen-Or, and B. Chen, "Active co-analysis of a set of shapes," *ACM Transactions on Graphics*, vol. 31, no. 6, pp. 1–10, 2012.
[45] H. Maron, M. Galun, N. Aigerman, M. Trope, N. Dym, E. Yumer, V. G. Kim, and Y. Lipman, "Convolutional neural networks on surfaces via seamless toric covers," *ACM Transactions on Graphics*, vol. 36, pp. 1–10, 2017.
[46] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "MeshCNN: A network with an edge," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–12, 2019.
[47] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov, "DiffusionNet: Discretization agnostic learning on surfaces," *ACM Transactions on Graphics*, vol. 41, pp. 1–16, 2020.
[48] T. W. Mitchel, V. G. Kim, and M. M. Kazhdan, "Field convolutions for surface CNNs," *IEEE/CVF International Conference on Computer Vision*, pp. 9981–9991, 2021.
[49] D. Smirnov and J. M. Solomon, "HodgeNet: Learning spectral geometry on triangle meshes," *ACM Transactions on Graphics*, vol. 40, pp. 166:1–166:11, 2021.
[50] A. Poulenard and M. Ovsjanikov, "Multi-directional geodesic neural networks via equivariant convolution," *ACM Transactions on Graphics*, vol. 37, pp. 1–14, 2018.
[51] Y. Yang, H. Pan, S. Liu, Y. Liu, and X. Tong, "PFCNN: Convolutional neural networks on 3D surfaces using parallel frames," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13 575–13 584, 2018.

Zhenyu Shu earned his PhD degree in 2010 at Zhejiang University, China. He is now working as a full professor at NingboTech University. His research interests include computer graphics, digital geometry processing, and machine learning. He has published over 40 papers in international conferences or journals.

Xufei Sun is a graduate student of the College of Computer Science and Technology at Zhejiang University. Her research interests include computer graphics and machine learning.

Chaoyi Pang is a senior member of ACM. He earned his PhD degree from the University of Melbourne (1999), advised by Prof. Gouzhu Dong and Prof. Rao Kotagiri. After receiving a PhD degree, he worked in the IT industry as an IT engineer and consultant in 1999-2002 and CSIRO as a research scientist in 2002-2014 in Australia. Currently, he is the Dean of the School of Computer and Data Engineering, Zhejiang University (NIT). He is a distinguished professor at Zhejiang University (NIT), Hebei Academy of Sciences, and Hebei University of Economics and Business. His research interests lie in algorithms, stream data compression and processing, data warehousing, data integration, database theory, graph theory and e-health.

Shiqing Xin is a full professor at the School of Computer Science and Technology at Shandong University. He received his PhD degree in applied mathematics at Zhejiang University in 2009. His research interests include computer graphics, computational geometry, and 3D printing.