Semi-supervised 3D Shape Segmentation via Self Refining

Zhenyu Shu, Teng Wu*, Jiajun Shen, Shiqing Xin, and Ligang Liu

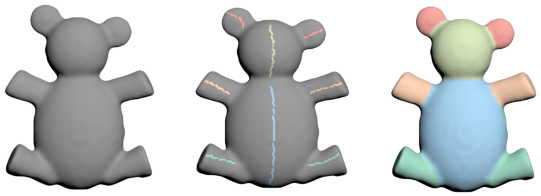
***Abstract*—3D shape segmentation is a fundamental and crucial task in the field of image processing and 3D shape analysis. To segment 3D shapes using data-driven methods, a fully labeled dataset is usually required. However, obtaining such a dataset can be a daunting task, as manual face-level labeling is both time-consuming and labor-intensive. In this paper, we present a semi-supervised framework for 3D shape segmentation that uses a small, fully labeled set of 3D shapes, as well as a weakly labeled set of 3D shapes with sparse scribble labels. Our framework first employs an auxiliary network to generate initial fully labeled segmentation labels for the sparsely labeled dataset, which helps in training the primary network. During training, the self-refine module uses increasingly accurate predictions of the primary network to improve the labels generated by the auxiliary network. Our proposed method achieves better segmentation performance than previous semi-supervised methods, as demonstrated by extensive benchmark tests, while also performing comparably to supervised methods.**

***Index Terms*—3D shape segmentation, Semi-Supervised, Deep neural network**

I. Introduction

SHAPE segmentation, which involves separating 3D shapes into meaningful parts, is crucial for efficiently processing 3D shapes. It enables the intrinsic properties of the shape, such as its structure, to be more easily understood. Consequently, various tasks such as mesh editing [1], reconstruction [2], [3], modeling [4], deformation [5], and shape retrieval [6], [7] rely on 3D shape segmentation to achieve satisfactory results. As a result, shape segmentation has become one of the most popular and challenging research fields.

The conventional approaches [8], [9], [10] to 3D shape segmentation typically involve three main steps. Firstly, hand-crafted shape descriptors are used to map each face on shapes to a corresponding feature vector. Subsequently, clustering or classification methods are applied in the feature space to assign a label to each feature vector. Finally, each face in the 3D shape is labeled based on the label of its corresponding feature vector. However, recent advances in machine learning have led to the development of learning-based segmentation methods [11], [12], [13], [14], particularly those based on deep learning architectures [15], [16], [17]. These methods have shown remarkable improvements in performance compared to traditional geometric optimization methods. Among these, projection techniques are utilized in 3D shape segmentation to obtain the multi-view rendered or depth images for each 3D shape using different camera settings while correspondences between faces and pixels are established [16], [18]. Semantic segmentation can be applied to the multi-view images to get labeled images, which are further back-projected to each 3D shape based on the established correspondences.



(a) Original 3D Shape (b) Scribble (c) Ground Truth

Fig. 1. (a) the original unlabeled 3D shape, (b) the 3D shape with sparse scribble labels, and (c) the fully labeled 3D shape.

Although learning-based segmentation methods, especially those using deep learning, have shown impressive results, they have a major drawback of requiring a vast amount of fully labeled training data that are similar to the target shape. This can be a significant burden in terms of the time and cost of manual labeling. As a result, these limitations of the existing learning-based segmentation methods have motivated the development of novel approaches that can overcome these issues.

To overcome the limitations of current learning-based segmentation methods, we propose a novel semi-supervised framework for 3D shape segmentation. In our approach, we divide the training dataset into two parts: a small amount of fully labeled 3D shapes (F) and a number of sparsely scribbled shapes (S). Figure 1 provides a visual comparison between an original 3D shape, a sparsely scribbled 3D shape, and a fully labeled 3D shape, highlighting the differences between them. Combined with our novel framework, our method can effectively reduce the workload of manual labeling, which are expensive and time-consuming.

Our framework consists of three modules. Firstly, the auxiliary model generates initial segmentation labels for sets with sparsely scribbled labels. Next, the primary segmentation model is trained with the support of the auxiliary model,

Zhenyu Shu is with School of Computer and Data Engineering, NingboTech University, Ningbo 315100, China. He is also with Ningbo Institute, Zhejiang University, Ningbo 315100, China (e-mail: shuzhenyu@nit.zju.edu.cn).

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

Jiajun Shen is with School of Software Technology, Zhejiang University, Ningbo 315048, China (e-mail: jiajunshen_paper@163.com).

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

Ligang Liu is with Graphics & Geometric Computing Laboratory, School of Mathematical Sciences, University of Science and Technology of China, Anhui 230026, China (e-mail: lgliu@ustc.edu.cn).

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

leveraging the initial labels to improve segmentation accuracy. Finally, the self-refine module utilizes increasingly accurate primary models to iteratively refine the generated labels during training.

Our contributions are as follows:

- We propose a semi-supervised framework for training 3D shape semantic segmentation. Our framework makes use of a small set of fully labeled 3D shapes and a set of sparsely scribbled labeled 3D shapes, significantly simplifying the 3D shape labeling process.
- We introduce two mechanisms in our framework: the auxiliary module generates label predictions at the face level for the scribble set, while the self-refine module uses a trainable CNN module to adjust the prediction results of both the primary and auxiliary modules on the scribble set.
- Extensive results from the public benchmark test showcase that our proposed approach outperforms previous unsupervised and semi-supervised methods in terms of semantic segmentation. Moreover, our method's performance is comparable to that of the fully supervised methods.

The rest of the paper is structured as follows. Section II provides a review of the related work. Section III presents a detailed explanation of our proposed method. In Section IV, we evaluate the performance of our algorithm on benchmark datasets. Section V discusses the limitations of our approach and suggests future research directions. Lastly, Section VI concludes the paper.

II. Related Work

3D shape segmentation is a fundamental and significant task in shape analysis and understanding as it aims to partition 3D shapes into meaningful parts that conform to human perception. In the past decade, there has been a surge in the development of novel and efficient 3D shape segmentation methods, primarily driven by advancements in machine learning, particularly deep learning techniques. Existing methods for 3D shape segmentation can be broadly classified into three categories: traditional 3D segmentation, unsupervised 3D shape segmentation, and deep learning-based 3D shape segmentation methods.

A. Traditional 3D Shape Segmentation

Early works in the field focus on designing suitable 3D shape feature descriptors that effectively capture the structural information. These methods involve extracting features from the 3D shape and utilizing classification algorithms to classify different regions. Commonly used features include shape descriptors such as surface normals and curvature, local features of voxels like local histograms, or histogram-based statistical features. Classification algorithms range from simple thresholding to more sophisticated techniques such as support vector machines and random forests [19], [20].

Region-growing-based methods [21] begin with known seed points and progressively expand and merge similar regions. These methods compute similarity measures, such as color, texture, or shape features, between neighboring regions to determine whether they should be merged.

Graph-based methods represent shapes as graphs, with nodes representing elements of the shape and edges representing their relationships. By defining an energy function on the graph and employing graph theory algorithms such as graph cuts (Boykov et al. [22]) or minimum spanning tree (Pettie et al. [23]) to minimize the energy function, shape segmentation can be achieved. In a specific study, Lai et al. [24] propose a fast and effective technique for segmenting 3D meshes into coherent regions using geometric and topological information.

B. Unsupervised 3D Shape Segmentation

Prior to benefiting from data-driven approaches, the research community treats 3D shape segmentation as a clustering challenge. Generally speaking, those unsupervised methods aim to partition the 3D shape into meaningful segments based on the similarity of their features. Techniques like k-means clustering, spectral clustering, or hierarchical clustering are commonly used to group similar elements together. Clustering algorithms rely on feature descriptors, such as geometric properties, surface normals, or local shape descriptors, to measure the similarity between elements.

Hu et al. [25] perform clustering to capture the inherent subspace structure within a collection of shapes, allowing for the discovery of shared patterns and variations. Kaick et al. [26] present a robust method for segmenting incomplete 3D point clouds into semantic parts by decomposing the shape into weakly convex components and merging similar components based on volumetric analysis. Lin et al. [27] propose to use the medial axis transform of 3D shapes to encode geometrical and structural information and obtain satisfactory segmentation results.

These methods have been widely used in the field of 3D shape segmentation in the past, with certain advantages and limitations. However, in recent years, deep learning methods have made significant progress in 3D shape segmentation, enabling better handling of complex shapes and large-scale data.

C. Deep Learning on 3D Shape Segmentation

The 3D shapes, represented by meshes, consist primarily of vertices, edges, and faces. Consequently, deep learning-based approaches for 3D shape segmentation revolve around these fundamental elements. Additionally, techniques employing multi-view projection often project the mesh onto 2D images to perform 3D shape segmentation tasks.

Guo et al. [28] and Yu et al. [29] employed transformers or recursive neural networks to assign labels to points on the surfaces of 3D shapes efficiently. MeshCNN [30] proposes an innovative neural network architecture tailored for deep learning on meshes. The network introduces a convolution operation known as "edge convolution" that directly operates on edges within the mesh. Moreover, MeshCNN incorporates "edge pooling", a pooling operation based on edge collapsing algorithms, to reduce computational load while preserving

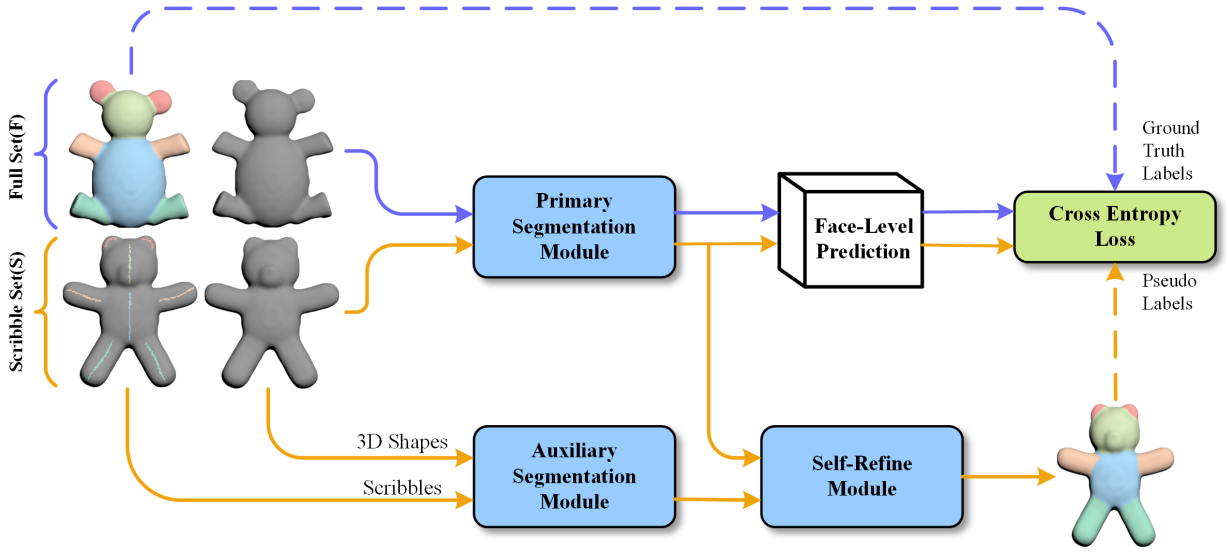


Fig. 2. Our segmentation framework comprises three components: i) The primary segmentation module: This module generates semantic segmentation predictions for a given 3D shape. It serves as the primary module for both training and testing. ii) The auxiliary segmentation module: This module outputs face-level segmentation predictions for each 3D shape with sparse scribble labels. It generates the initial segmentation for the scribble set, which is then used as input for training the primary network. iii) The self-refine module: This module refines the segmentation predictions generated by the auxiliary network and the current primary network for the scribble sets. It improves the accuracy of the segmentation results. The primary network is trained using cross-entropy loss, where its output is matched with the ground-truth segmentation label of the fully labeled shape or the output of the self-refine module for the scribble set. This training process ensures that the primary network learns to produce accurate segmentation predictions.

high resolution. Lahav et al. [31] utilize random walk algorithms to learn the global structure and local properties of meshes. Inspired by [32], Milano et al. [33] combine convolution operations on both primal and dual meshes and introduce a novel distance metric to enhance learning efficiency and accuracy. HodgeNet [34] learns a sparse differential operator parameterized using discrete exterior calculus, computes its low-order eigendecomposition, and produces per-vertex features using the spectral geometry. By transforming input meshes into the Laplacian-Beltrami spectral domain, Laplacian2Mesh [35] enables the use of mature CNN architectures for shape analysis tasks without dealing with irregular mesh connectivity.

Apart from the utilization on vertices and edges, several approaches deploy neural networks on the faces of meshes. Kalogerakis et al. [11] present a data-driven approach to simultaneously segment and label parts of 3D meshes using a Conditional Random Field model learned from labeled training meshes. Guo et al. [15] propose a 3D mesh labeling method using deep convolutional neural networks to learn robust mesh representations from geometric features and generate label vectors for triangles. Several recent works [36], [37], [38], [39], [40] have focused on developing deep neural network frameworks for learning 3D shape representation from mesh data.

In most of the methods mentioned earlier, the geometric features of 3D shapes, such as feature descriptors of faces, dihedral angles of edges, etc., are used as training data. However, multi-view-based methods solve the segmentation task of 3D shapes by establishing a mapping relationship between the 3D shape and its projected images. Wang et al. [18] and Kalogerakis et al. [16] employed projection matrices to project 3D shapes onto 2D images, followed by the application of image-based segmentation algorithms for achieving segmentation of the 3D shapes. Concurrently, Several recent works [41], [42], [43] have been a growing interest in methods for capturing features of 3D meshes using multi-view approaches.

Supervised learning methods require a large amount of fully labeled 3D shapes as training datasets. However, manually labeling a significant number of 3D shapes is a time-consuming and labor-intensive task. To mitigate this problem, several semi-supervised and weakly supervised methods are proposed. Shu et al. [44] propose a novel weakly-supervised 3D shape segmentation method that relies only on sparse scribble-based labels for training and achieves comparable performance to supervised methods. Zhuang et al. [45], [46] develope two semi-automated approaches to decompose a mesh into multiple patches with boundaries corresponding to ridge and valley lines. Tao et al. [47] employ an interactive weak labeling approach to indicate each instance's location in point cloud scenes precisely. Another semi-supervised 3D shape segmentation method is provided by Shu et al. [48], which combines soft density peak clustering and an optimization model for propagating labels to unlabeled parts.

To overcome the same limitations as the prior semi-supervised and weakly supervised methods, we propose a novel semi-supervised framework for 3D shape segmentation that utilizes only a small number of 3D shapes with patch-level semantic segmentation labels and another group of 3D shapes with sparse scribble labels. Our framework begins by training an auxiliary network to generate initial face-level label predictions for 3D shapes in the sparsely-labeled dataset S. During the training of the primary model, the self-refine module improves the labels of dataset S, resulting in excellent segmentation results while greatly simplifying the labeling process for 3D shapes.

III. Our Method

This section begins with an overview of the algorithm's pipeline, followed by individual introductions to each module within the framework. Finally, the complete training process is presented.

A. Overview

Our method employs two training sets to train the semi-supervised semantic segmentation network: 1) a small dataset F with fully labeled data, and 2) a large dataset S comprising sparsely scribble-labeled data. Our overall pipeline, illustrated in Figure 2, consists of three modules: i) The primary segmentation network generates face-level semantic segmentation predictions for a given unlabeled 3D shape. ii) The auxiliary segmentation network generates face-level semantic segmentation predictions for 3D shapes that only have sparse scribble labels. This module generates the initial face-level segmentation for the dataset S, which serves as input to aid in training the primary network. iii) The self-refine module refines the segmentation prediction results of the dataset S, generated by both the auxiliary network and the current primary network.

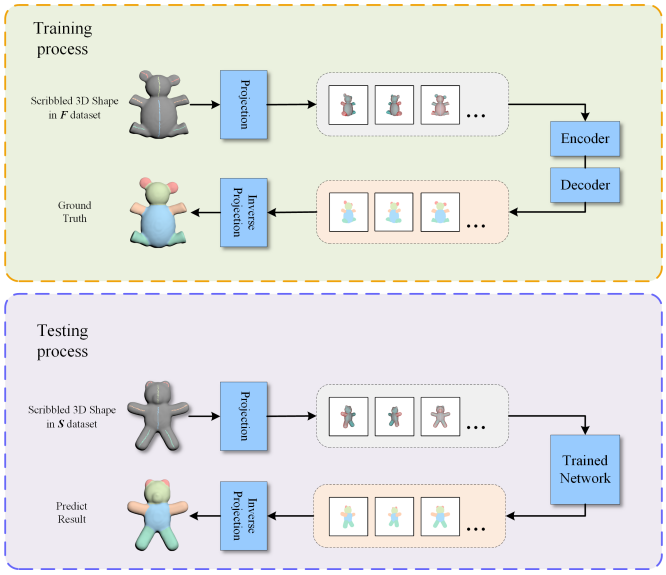


Fig. 3. For each sparsely labeled 3D shape, we project the shape using different camera parameter settings to obtain depth maps and rendered images. Subsequently, we transform these two types of single-channel 2D images into three-channel images, and we also establish correspondence between 2D images and faces on 3D shapes. Based on these combined images, we conduct training for image semantic segmentation. The auxiliary network is trained using dataset F, and the trained model is applied to dataset S to obtain face-level labels for dataset S.

B. Primary Segmentation Module

By excluding the self-refine and auxiliary modules, the primary segmentation module can be trained using ground truth labels on the fully supervised set F, employing the cross-entropy loss function:

$$\mathcal{L}_f = -\frac{1}{|F|}\sum_{f\in F}\sum_{y} y \log p_{pri}\left(y|\mathbf{x};\phi\right), \tag{1}$$

where $\mathbf{x}$ represents the feature vector of the face $f \in F$, while p_{pri} represents the prediction of the primary network for the input $\mathbf{x}$.

In our network, we use five established geometric feature descriptors, including AGD [49], GC [50], SDF [51], SIHKS [52], and WKS [53], to extract the geometric feature vectors of each face, whose dimensions are 1, 1, 1, 19, and 100 respectively. We concatenate the five feature vectors into one 122-dimensional vector $\mathbf{x}^{(f)}$. In the implementation, the primary network is composed of five fully connected layers, where the number of neurons of each layer is 122, 60, 40, 10, and C (the number of segments), respectively.

However, the fully supervised training method demands a substantial quantity of fully labeled data. The effectiveness of the model's training diminishes when the training dataset is relatively small, such as when utilizing solely the dataset F. Nevertheless, manual labeling of additional 3D shapes is a laborious and time-consuming task. Consequently, we introduce an auxiliary module that significantly simplifies the annotation process by solely relying on sparsely labeled 3D shapes as input. This module empowers us to acquire face-level semantic segmentation predictions for these 3D shapes.

C. Auxiliary Segmentation Module

When training a 3D shape semantic segmentation network using sparse scribble labels, propagating the labels poses a significant challenge. Existing methods primarily depend on manually designed rule-based processes, such as representing the 3D shape as a graph and employing graph-cut algorithms for label propagation. Additionally, some researchers have investigated iterative segmentation and sparse label propagation of 3D shapes by devising suitable similarity matrices.

In contrast, our approach uses a multi-view projection model to project the 3D shape into 2D images. For each sparsely scribble-labeled 3D shape, we define 32 virtual cameras at different positions placed at the bounding sphere radius of the shape. Additionally, each camera is rotated four times at 90-degree intervals. Therefore, for each input 3D shape, these cameras generate 128 sets of depth maps and rendered images, which can cover almost all vertices and facets of each shape in the dataset used in our experiments. At the same time, we establish a reference matrix to record the correspondence between 2D images and faces on 3D shapes. We transform the single-channel depth images and rendered images from each set of projected images into three-channel images. In this process, the positive depth images are designated as the first channel, the rendered images are the second channel and the negative depth images are set as the third channel. Next, we use these combined images as input to train the DeepLabv3+ image semantic segmentation network in an end-to-end manner, generating label predictions for these images and mapping the pixel label predictions back to the faces of the 3D shape using the established mapping relationship. As a result, label probability distribution results are generated for

each face of the input 3D shape (as illustrated in Figure 3). Benefiting from the pre-trained image semantic segmentation network on a large-scale image dataset, we achieve a more efficient and accurate label propagation process for the input sparsely labeled 3D shape.

This module is trained using dataset F and can serve as a 3D shape-label prediction model for dataset S. The auxiliary module is trained on the fully supervised set F using cross-entropy loss:

$$\mathcal{L}_{aux} = -\frac{1}{|F|}\sum_{f\in F}\sum_{y} y \log p_{aux}\left(y|\mathbf{x}, s; \boldsymbol{\theta}\right), \tag{2}$$

where s denotes the sparse scribble labels obtained by randomly sampling the ground truth labels of the 3D shapes in dataset F, and p_{aux} represents the prediction of the auxiliary network for the input $\mathbf{x}$.

In subsequent experiments, the model parameters $\boldsymbol{\theta}$ remain unchanged. During the prediction process, 3D shapes labeled with sparse scribbles from dataset S are input into the network to obtain the prediction of face-level semantic segmentation labels. After adding the auxiliary module, when processing 3D shapes in dataset S with sparse scribble labels, the primary network first uses soft labels generated by the auxiliary network for these shapes and then trains with the cross-entropy loss:

$$\mathcal{L}_{s} = -\frac{1}{|S|}\sum_{f\in S}\sum_{y} p_{aux}\left(y|\mathbf{x}, s; \boldsymbol{\theta}\right) \log p_{pri}\left(y|\mathbf{x}; \boldsymbol{\phi}\right), \tag{3}$$

where $\mathbf{x}$ represents the feature vector of face $f \in S$, p_{aux} is the soft label generated by the auxiliary network for the input $\mathbf{x}$ with sparse scribble s, and p_{pri} represents the prediction of the primary network for the input $\mathbf{x}$. $\mathcal{L}_s$ denotes the cross-entropy loss targeting the soft labels generated by the auxiliary network.

Without the self-refine module, the primary network is trained by using the ground truth labels obtained from dataset F and the labels generated from the auxiliary network for dataset S. The overall loss function for this training process can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_f + \mathcal{L}_s. \tag{4}$$

D. Convolutional Self-Refine Module

The equations mentioned above rely on the auxiliary model to predict the label distribution for the face-level segmentation of 3D shapes in the dataset S. However, the pseudo-labels generated by the auxiliary model possess limited accuracy. In the early stages of training the primary network, these rough pseudo-labels can be beneficial. However, in the later stages of training, they lose their effectiveness and may even have a harmful impact on the training of the primary network. This is primarily due to the introduction of erroneous labels, which can cause the network to converge in an incorrect direction.

Therefore, we propose a novel convolutional self-refine module that combines the predictions of both the primary and auxiliary networks to generate face-level labels to 3D shapes in dataset S. The proposed module comprises two layers of 5x5 convolutional layers, which learn the weights

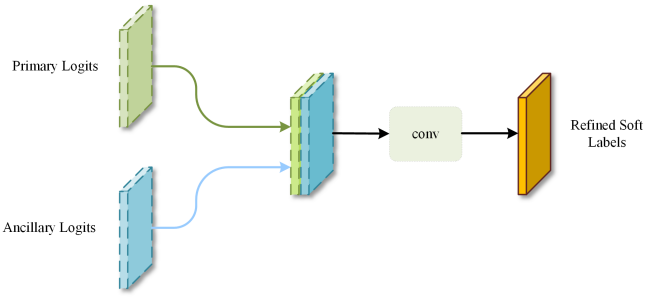


Fig. 4. The convolutional self-refine module refines the label predictions of the auxiliary network for the 3D shapes in dataset S. This network takes the predictions of the primary network and the auxiliary network on dataset S as input and combines their outputs. The combined output is then fed into a four-layer CNN module.

of predictions from primary and auxiliary networks during the training process. By incorporating information from both networks, the self-refine module refines the label predictions and enhances the accuracy of the final results. As illustrated in Figure 4, during the training of the primary network, the network can dynamically adjust the weights assigned to the predictions on dataset S. In case the primary model offers more accurate label predictions, the module assigns a higher weight to the primary model's predictions. This adaptive weighting mechanism enables the module to leverage the strengths of both models and significantly enhance the overall performance.

To ensure that the self-refine module can provide more accurate predictions than the auxiliary or primary networks, we introduce an $\mathcal{L}_{conv}$ term in the total loss function. This term enables the self-refine module to learn from dataset F while concurrently training the primary network on the entire dataset:

$$\mathcal{L}_{conv} = -\frac{1}{|F|}\sum_{f\in F}\sum_{y} y \log P_{conv}, \tag{5}$$

where P_{conv} is defined as follow:

$$P_{conv} = p_{conv}\big(y|p_{pri}(\mathbf{x}, \boldsymbol{\phi}), p_{aux}(\mathbf{x}, s, \boldsymbol{\theta}); \boldsymbol{\lambda}\big). \tag{6}$$

The convolutional self-refine network takes the predicted logits generated by the auxiliary and primary networks and generates the final segmentation soft labels p_{conv}. Here, $\boldsymbol{\lambda}$ represents the parameters of the convolutional self-refine network.

With the introduction of the convolutional self-refine module for the 3D shapes in dataset S, the training cross-entropy loss function of the primary network will be updated as follows:

$$\mathcal{L}_{s} = -\frac{1}{|S|}\sum_{f\in S}\sum_{y} P_{conv} \log p_{pri}\left(y|\mathbf{x}; \boldsymbol{\phi}\right). \tag{7}$$

E. Training Process.

The total loss function for our proposed method is defined as follows:

$$\mathcal{L} = \mathcal{L}_f + \mathcal{L}_s + \mathcal{L}_{conv}. \tag{8}$$

In the course of our experimentation, we observed that pre-training the auxiliary network on the entire dataset F resulted

ALGORITHM 1: Training process of our method

Input:
3D shapes within fully labeled dataset F and sparsely scribble-labeled dataset S.
Output:
Trained primary network for 3D shape segmentation.
Training process:
Step 1: Initial training of the auxiliary network;
for $f \in F/2$ **do**
Project face f with scribble s into pixel;
Obtain auxiliary prediction $p_{aux}(y|\mathbf{x}, s; \boldsymbol{\theta})$;
Update loss $\mathcal{L}_{aux}$ based on Eq. (2).
Step 2: Initial training of self-refine network;
for $f \in F$ **do**
Calculate primary prediction $p_{pri}(y|\mathbf{x}; \boldsymbol{\phi})$;
Obtain auxiliary prediction $p_{aux}(y|\mathbf{x}, s; \boldsymbol{\theta})$;
Get self-refine labels $p_{conv}(y|p_{pri}, p_{aux}; \boldsymbol{\lambda})$;
Update loss $\mathcal{L}_f$ and $\mathcal{L}_{conv}$ based on Eq. (1, 5).
Step 3: Training primary network.
for $f \in F \cup S$ **do**
Calculate primary prediction $p_{pri}(y|\mathbf{x}; \boldsymbol{\phi})$;
Obtain auxiliary prediction $p_{aux}(y|\mathbf{x}, s; \boldsymbol{\theta})$;
Get self-refine labels $p_{conv}(y|p_{pri}, p_{aux}; \boldsymbol{\lambda})$;
Update total loss $\mathcal{L}$ based on Eq. (8).

in highly precise predictions for the dataset F. In such cases, the subsequent convolutional self-refine network becomes overly rely on the predictions of the auxiliary network, failing to effectively learn how to balance the predictions of both the auxiliary and primary networks. To clarify our method, Algorithm 1 presents the training process of our approach.

IV. Experiments

This section first provides a comprehensive overview of our experiments' evaluation metrics, implementation details, and training dataset splitting strategies. Then, we present the qualitative and quantitative results of our method on the widely used PSB, COSEG, ShapeNetCore, and Human Body datasets. Moreover, we compare our method with the results of existing state-of-the-art 3D shape segmentation methods. We also conducted some ablation studies to verify the effectiveness of components in our method.

Evaluation Metrics. In comparison with the supervised learning methods such as [11] and [15], we measure the performance using accuracy, as follows:

$$Accuracy = \sum_{i \in T} t_i \mathbf{u}\left(l_i\right) / \sum_{i \in T} t_i. \tag{9}$$

Additionally, as some existing methods do not provide accuracy results and instead use the Rand Index as an evaluation metric, we have also employed the Rand Index to compare our method with these approaches. The Rand Index proposed by [54] is a widely used and comprehensive evaluation metric for assessing the differences between two segmentation results. Typically, it is computed by comparing the segmentation results with manually labeled ground truth. A lower Rand Index indicates a more significant similarity between two segmentations, implying better segmentation performance of the algorithm.

Implementation Details. We use the public Matlab implementation of DeepLabv3+ as the foundation for our auxiliary model, which is trained using a batch size of 16 and an initial learning rate of 0.005. Our primary network is implemented based on MLP, which is trained with a batch size of 32 and an initial learning rate of 0.001.

A. Experimental Datasets

In this section, we evaluate the performance of our method and conduct a comprehensive comparison with various other 3D shape segmentation methods. For this purpose, we have selected four benchmark datasets: PSB, COSEG, ShapeNetCore, and Human Body.

The PSB dataset, initially presented by Chen et al. [54], has 19 categories, each including 20 3D shapes. To ensure an unbiased evaluation, we employ ground truth segmentation labels provided by Kalogerakis et al. [11] as a standard for evaluation. The COSEG dataset [55] is composed of two parts: a small dataset with 190 shapes from 8 categories and a larger dataset with 200, 400, and 300 shapes from the categories of Tele-aliens, chairs, and vases, correspondingly. Similar to the PSB dataset, most shapes in COSEG have undergone preprocessing, leading to a topology well-suited for geometric processing applications. The ShapeNetCore dataset is a subset of the ShapeNet dataset described in [56]. To address the issue of non-manifold shapes, we utilized a technique proposed by [57] that converts such shapes into manifold ones. This approach has been shown to improve the processing and analysis of the dataset. The Human Body dataset [58] consists of 370 training models from SCAPE, FAUST, MIT, and Adobe Fuse and 18 testing models from the SHREC07 (humans) dataset. These models were manually segmented into eight labels corresponding to the labels provided by Kalogerakis et al. [11].

B. Training Dataset Splitting

In each dataset category, when comparing with other fully supervised methods, our approach employs two training data splitting strategies: one combines 20% fully labeled shapes with 40% scribble-based partially labeled shapes as training data (1+2+2). The other combines 40% fully labeled shapes with 20% scribble-based partially labeled shapes as training data (2+1+2). The remaining 40% of shapes are designated as the testing data. In contrast, all other fully supervised methods in our comparisons use 60% fully labeled shapes as their training data. Compared with other semi-supervised methods, our approach uses 20% fully labeled shapes and 20% scribble-based partially labeled shapes as training data, randomly selecting 40% of the remaining shapes as testing data. Meanwhile, all other semi-supervised methods in our comparisons use 40% fully labeled shapes and 20% unlabeled shapes as their training data. It is noteworthy that all the subsets used for training are randomly split.

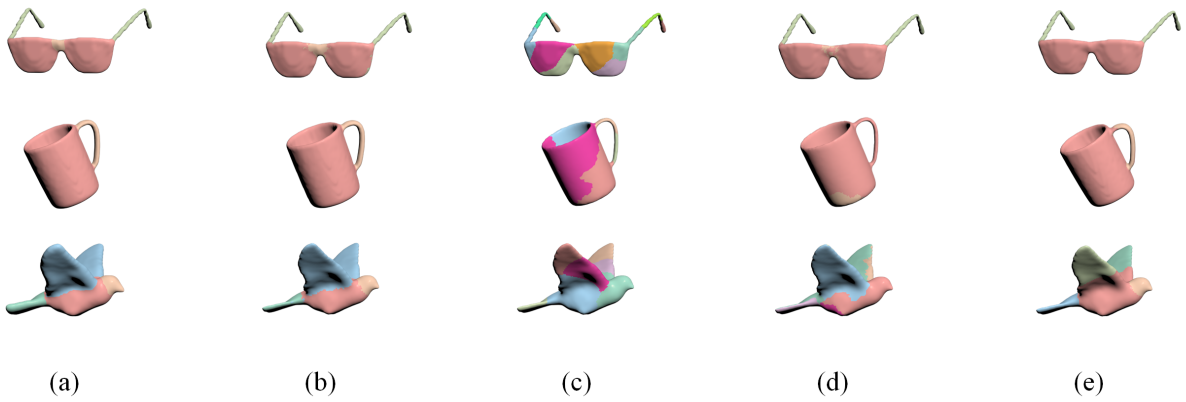


Fig. 5. Comparison with other three segmentation algorithms. The methods displayed are as follows: (a) Ground Truth, (b) Our method, (c) RandomWalks, (d) WeaklyScribble, and (e) SEMI3.

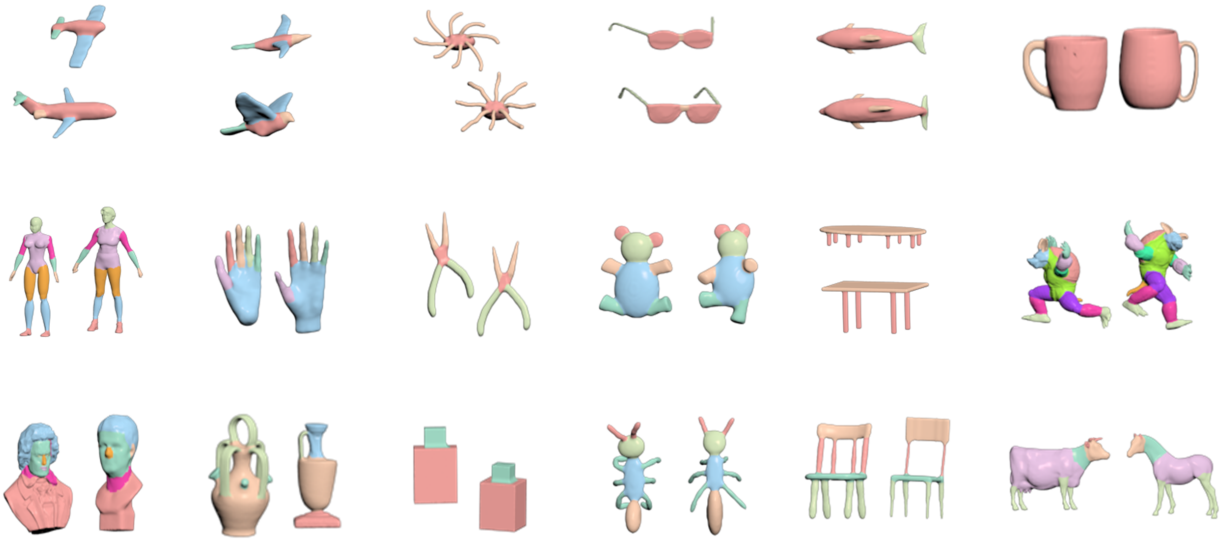


Fig. 6. Several experimental results showcasing the performance of our method on the PSB dataset.

C. Experimental Results and Comparisons

In our work, we evaluated the effectiveness of our proposed method by comparing it with various unsupervised, weakly supervised, and semi-supervised methods on the PSB dataset., and conducted comparisons with fully supervised methods across all datasets. To ensure a fair comparison, we extensively referred to the results reported in the original papers of the methods we compared against.

Comparison with unsupervised, weakly supervised, and semi-supervised methods. We compared our method with RandomWalks [24] and WcSeg [26], two unsupervised methods, and a weakly supervised method named WeaklyScribble [44]. We also compared with three semi-supervised methods, including SEMI-1 [45], SEMI-2 [46], and SEMI-3 [48]. As shown in Table I, we compare our method with these methods in terms of Rand Index scores on the PSB dataset. Our experimental results demonstrate that our method outperforms other unsupervised, weakly supervised, and semi-supervised methods in 9 out of 16 categories, achieving the lowest Rand Index scores. Additionally, our proposed semi-supervised approach achieved an average Rand Index score of 0.53, surpassing all other methods compared. Figure 5 shows a qualitative comparison of our method with unsupervised, weakly supervised, and semi-supervised methods on the Glasses, Cup, and Bird models. Our experimental results demonstrate that our proposed semi-supervised method exhibits superior segmentation performance compared to other existing methods. This notable achievement can be primarily attributed to incorporating our self-refine module, which iteratively enhances the precision of labels during the training process.

Comparison with supervised methods. On the PSB dataset, we conducted comparisons with seven other fully supervised methods, including ShapeBoost [11], Hu et al. [25], Wang et al. [18], Guo et al. [15], ShapePFCN [16], MeshCNN [30], and MeshWalker [31]. Based on the data in Table II, our method achieved higher accuracy results in 8 categories than

TABLE I
EVALUATING THE PERFORMANCE OF OUR METHOD AND COMPARING IT WITH OTHER UNSUPERVISED, WEAKLY SUPERVISED, AND SEMI-SUPERVISED METHODS USING RAND INDEX SCORES ON THE PSB DATASET.

Methods	RandomWalks [24]	WcSeg [26]	WeaklyScribble [44]	SEMI-1 [45]	SEMI-2 [46]	SEMI-3 [48]	Ours
Human	0.363	0.128	**0.064**	0.185	0.162	0.119	0.079
Cup	0.311	0.165	0.021	0.103	0.099	0.036	**0.019**
Glasses	0.395	0.175	0.048	0.181	0.176	0.150	**0.036**
Airplane	0.283	0.089	0.081	0.121	0.103	**0.029**	0.055
Ant	0.092	0.021	**0.007**	0.050	0.034	0.016	0.011
Chair	0.202	0.105	0.029	0.094	0.079	0.059	**0.028**
Octopus	0.135	0.029	0.022	0.051	0.039	**0.019**	0.020
Table	0.141	0.089	0.014	0.074	0.070	0.035	**0.012**
Teddy	0.116	0.057	0.068	0.056	**0.045**	0.050	0.054
Hand	0.236	0.112	0.078	0.118	0.104	0.054	**0.048**
Plier	0.277	0.086	0.050	0.103	0.094	0.087	**0.045**
Fish	0.381	0.203	**0.100**	0.196	0.178	0.118	0.112
Bird	0.250	0.103	0.092	0.098	0.094	0.050	**0.030**
Armadillo	0.200	0.080	0.065	0.132	0.118	0.098	**0.059**
Vase	0.253	0.162	**0.101**	0.184	0.165	0.141	0.112
FourLeg	0.431	0.153	0.148	0.167	0.134	0.150	**0.129**
Average	0.259	0.121	0.062	0.120	0.106	0.076	**0.053**

TABLE II
COMPARING OUR METHOD WITH OTHER SUPERVISED METHODS ON THE PSB DATASET.

Methods	ShapeBoost [11]	Hu [25]	[18]	[15]	ShapePFCN [16]	MeshCNN [30]	MeshWalker [31]	Ours (1+2+2)	Ours (2+1+2)
Human	93.20%	70.40%	55.60%	91.22%	93.80%	74.76%	87.02%	92.80%	**95.35%**
Cup	99.60%	97.40%	99.60%	**99.73%**	93.70%	95.86%	99.54%	97.53%	99.10%
Glasses	97.20%	**98.30%**	-	97.60%	96.30%	93.94%	96.11%	96.20%	97.46%
Airplane	96.10%	83.30%	-	96.67%	92.50%	84.36%	96.20%	95.39%	**97.15%**
Ant	98.80%	92.90%	-	98.80%	**98.90%**	91.83%	97.36%	98.17%	98.55%
Chair	98.40%	89.60%	**99.60%**	98.67%	98.10%	84.75%	97.61%	98.78%	99.30%
Octopus	98.40%	97.50%	-	98.79%	98.10%	98.21%	97.86%	98.80%	**99.02%**
Table	99.30%	99.00%	**99.60%**	99.55%	99.30%	96.78%	99.33%	99.10%	**99.47%**
Teddy	98.10%	97.10%	-	98.24%	96.50%	84.29%	95.57%	98.55%	**98.88%**
Hand	88.70%	**91.90%**	-	88.71%	88.70%	68.83%	83.31%	82.88%	87.85%
Plier	96.20%	86.00%	-	96.22%	95.70%	83.69%	92.24%	**96.64%**	**96.64%**
Fish	95.60%	85.60%	-	95.64%	95.90%	89.05%	94.58%	95.79%	**96.11%**
Bird	87.90%	71.50%	-	88.35%	86.30%	68.09%	**92.76%**	89.44%	90.88%
Armadillo	90.10%	87.30%	-	92.27%	**93.30%**	50.24%	89.12%	90.01%	92.97%
Vase	85.80%	80.20%	**90.50%**	89.11%	85.70%	68.94%	84.56%	83.22%	88.52%
FourLeg	86.20%	88.70%	54.30%	87.02%	89.50%	68.73%	80.93%	84.32%	**90.33%**
Average	94.35%	88.54%	-	94.79%	93.89%	81.40%	92.76%	94.10%	**96.10%**

TABLE III
COMPARING OUR METHOD WITH OTHER SUPERVISED METHODS ON THE SMALL COSEG DATASET.

Methods	[15]	ShapePFCN [16]	MeshCNN [30]	Ours (1+2+2)	Ours (2+1+2)
Candelabra	85.90%	**95.40%**	83.52%	87.32%	93.28%
Chairs	93.80%	96.10%	92.87%	95.24%	**97.05%**
Fourleg	88.20%	90.40%	86.19%	90.48%	**92.10%**
Goblets	86.10%	**97.20%**	92.62%	93.76%	95.60%
Guitars	97.70%	98.00%	91.34%	98.15%	**98.45%**
Irons	79.70%	88.00%	81.26%	88.30%	**90.39%**
Lamps	78.00%	**93.00%**	83.64%	86.57%	90.15%
Vases	84.40%	84.80%	77.43%	85.60%	**86.88%**
Average	86.72%	92.86%	86.11%	91.93%	**92.99%**

other fully supervised algorithms and attained the best average accuracy results. Figure 6 displays the qualitative results of our algorithm on the PSB dataset, illustrating the segmentation capabilities of our method when handling various 3D shapes. Specifically, both our method and ShapePFCN utilize multi-view projection techniques. However, our proposed semi-supervised method leverages information from both 2D and 3D levels to effectively address the challenges associated with geometric topological information loss and shape occlusion in the multi-view projection process. This approach leads to a significant reduction in the incidence of misclassification. Figure 7 shows an example of a qualitative comparison between our method and ShapePFCN on the FourLeg category of the PSB dataset.

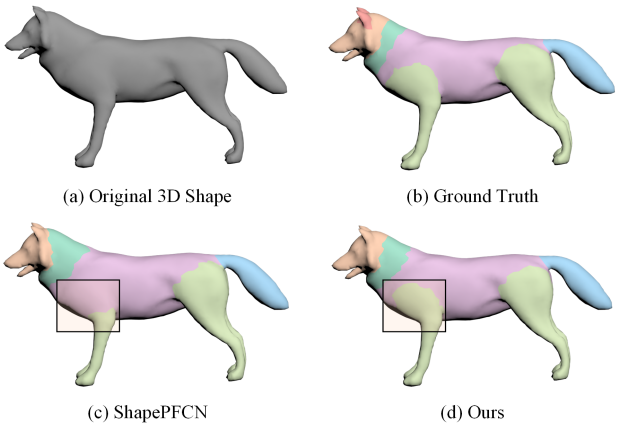
On the small COSEG dataset, we also compared our method with three fully supervised methods, including Guo et al., ShapePFCN, and MeshCNN, as detailed in Table III. Similarly, on the large COSEG dataset, our method was compared against seven fully supervised methods, namely MeshCNN, MeshWalker, PDMeshNet [33], HodgeNet [34], SubdivNet [39], Laplacian2Mesh [35], and DGNet [40], with specific results in Table IV. Our experimental results demon-

TABLE IV
COMPARING OUR METHOD WITH OTHER SUPERVISED METHODS ON THE LARGE COSEG DATASET.

Methods	MeshCNN [30]	MeshWalker [31]	PDMeshNet [33]	HodgeNet [34]	SubdivNet [39]	Laplacian2Mesh [35]	DGNet [40]	Ours (1+2+2)	Ours (2+1+2)
Tele-aliens	95.76%	**98.70%**	98.18%	96.03%	97.30%	95.00%	97.40%	97.82%	98.55%
Chairs	94.54%	98.60%	97.23%	95.68%	96.70%	96.60%	96.70%	97.96%	**99.05%**
Vases	93.49%	**99.90%**	95.36%	90.30%	96.70%	94.60%	97.00%	96.36%	97.88%
Average	94.60%	**98.77%**	96.92%	94.00%	96.90%	95.40%	97.03%	97.38%	98.49%

TABLE V
COMPARING OUR METHOD WITH OTHER SUPERVISED METHODS ON THE SHAPENETCORE DATASET. WE USE THE ACCURACY (%) METRIC TO COMPARE OUR METHOD WITH MESH-BASED METHODS AND THE MIOU (%) METRIC TO COMPARE WITH POINT-BASED METHODS.

Algorithms	aero	bag	cap	car	chair	eph.	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate.	table	mean
ShapeBoost [11]	85.8	93.1	85.9	79.5	70.1	81.4	89.0	81.2	71.7	86.1	77.2	94.9	88.2	79.2	91.0	74.5	83.0
[15]	87.4	91.0	85.7	80.1	66.8	79.8	89.9	77.1	71.6	82.7	80.1	95.1	84.1	76.9	89.6	77.8	82.9
ShapePFCN [16]	90.3	94.6	**94.5**	**86.7**	**82.9**	84.9	91.8	82.8	**78.0**	**95.3**	**87.0**	96.0	91.5	81.6	91.9	**84.8**	88.4
SEG-MAT [59]	83.7	-	-	-	80.3	82.1	90.9	83.2	-	-	-	-	-	74.3	79.6	80.4	81.8
Ours (1+2+2)	**92.0**	**96.1**	92.8	84.1	81.4	**86.4**	**93.2**	**84.3**	76.4	94.6	85.1	**97.5**	**93.3**	**83.5**	**93.8**	83.0	**88.6**
PartNet [29]	**87.8**	86.7	**89.7**	80.5	91.9	75.7	**91.8**	85.9	83.6	**97.0**	74.6	**97.3**	83.6	64.6	**78.4**	**85.8**	87.4
PCT [28]	85.0	82.4	89.0	81.2	91.9	71.5	91.3	**88.1**	86.3	95.8	64.6	95.8	83.6	62.2	77.6	83.7	83.1
Point-BERT [60]	84.3	84.8	88.0	79.8	91.0	81.7	91.6	87.9	85.2	95.6	75.6	94.7	84.3	63.4	76.3	81.5	84.1
Ours (1+2+2)	87.4	**91.3**	89.5	**82.5**	**92.3**	**82.0**	91.5	88.0	**88.4**	96.5	**82.3**	97.1	**89.0**	**66.4**	77.2	84.6	**87.8**



(a) Original 3D Shape (b) Ground Truth (c) ShapePFCN (d) Ours

Fig. 7. Qualitative results for the FourLeg category of the PSB dataset. In comparison with ShapePFCN, our method avoids mis-classifications.

Fig. 8. Several experimental results demonstrating the performance of our method on the COSEG small dataset.

strate that our method outperforms all other compared methods on the small COSEG dataset, with an average accuracy rate of 92.99%. However, on the large COSEG dataset, while MeshWalker, a fully supervised method, outperforms our proposed semi-supervised method, our method still achieves the highest segmentation accuracy in the Chairs category. Notably, our approach used less supervisory information and achieved an average accuracy rate that was only 0.28% lower than MeshWalker's accuracy rate. Moreover, our results on the PSB dataset outperform MeshWalker. We attribute our better performance on the PSB dataset to the quality of scribble annotations we employed on the PSB dataset. Furthermore, our method outperformed all other fully supervised methods in the comparison. Figures 8 and 9 showcase the qualitative results of our method on the COSEG dataset.

In addition, to further evaluate the performance of our method, we conducted experiments on the ShapeNetCore and Human Body datasets. On the ShapeNetCore dataset, we compared our method with seven fully supervised methods, including four mesh-based methods: ShapeBoost, Guo's method, ShapePFCN, and SEG-MAT [59], as well as point-based methods: PartNet [29], PCT [28], and Point-BERT [60]. The detailed results of these comparisons can be found in Table V. We employed the accuracy(%) evaluation metric when comparing with mesh-based methods and the mIoU(%)

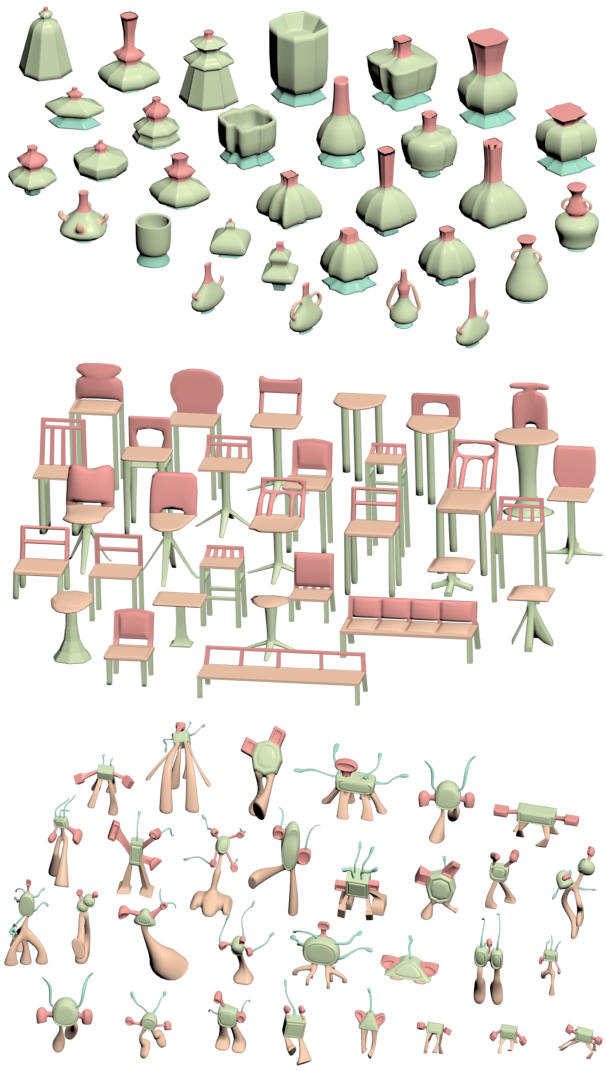


Fig. 9. Some experimental results of our method on the COSEG large dataset.

TABLE VI
THE ACCURACY OF SEGMENTATION ON THE HUMAN BODY DATASET COMPARED WITH OTHER SUPERVISED METHODS.

Method	Accuracy	Method	Accuracy
[58]	88.00%	MDGCNN [61]	88.61%
PFCNN [62]	91.45%	HodgeNet [34]	85.03%
SubdivNet [39]	91.70%	DiffusionNet [63]	90.80%
Ours	**93.15%**		

evaluation metric when comparing with point-based methods. Similarly, on the Human Body dataset, we compared our method with fully supervised methods such as Maron et al. [58], MDGCNN [61], PFCNN [62], HodgeNet, SubdivNet, and DiffusionNet [63]. The comparative results are shown in Table VI. We achieve the highest average evaluation scores on the ShapeNetCore and Human Body datasets, and Figure 10 displays the segmentation results of our algorithm on the ShapeNetCore and the Human Body dataset.

To further validate the efficacy and stability of our proposed method, we conducted comparative experiments on the

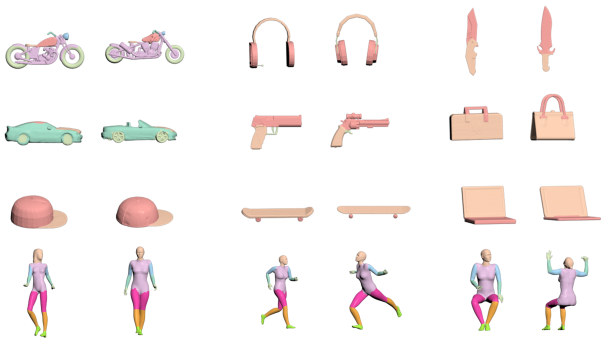


Fig. 10. Several experimental results demonstrating the performance of our method on the ShapeNetCore and the Human Body dataset.

TABLE VII
COMPARISON OF AVERAGE ACCURACY RESULTS UNDER DIFFERENT TRAINING DATASET SIZES.

Method	30%	50%	80%	100%
MeshCNN [30]	75.07%	87.22%	89.21%	90.84%
MeshWalker [31]	70.00%	72.03%	87.02%	91.37%
Ours	**90.04%**	**92.36%**	**93.33%**	**93.52%**

Human Body dataset, selecting two fully supervised methods, MeshCNN and MeshWalker, as baselines. Using the complete training dataset as a standard, we trained MeshCNN, MeshWalker, and our method with 30%, 50%, 80%, and 100% of the complete training dataset as the actual training dataset. Specifically, for our method, we divided the training dataset into two parts: two-thirds as dataset F and the remaining third using only the corresponding scribble labels as dataset S. Subsequently, we tested the shapes on the complete test dataset provided by the Human Body dataset. As shown in Table VII, the average accuracy of MeshCNN and MeshWalker methods on the test dataset significantly decreased with the reduction of fully labeled training data. In contrast, our proposed semi-supervised framework maintained an average accuracy of about 90%. This result demonstrates that our proposed semi-supervised framework is more stable with the reduction of fully labeled data and can still achieve excellent segmentation performance with less training data.

D. Ablation Study

In this section, we conducted ablation experiments on the critical components of our algorithm.

1) Network Architecture: To validate the contributions of each component within our proposed semi-supervised framework to the final segmentation network's effectiveness, we conducted an in-depth analysis to evaluate the influence of different components on segmentation performance. As illustrated in Figure 11, we performed comprehensive evaluations on four datasets: PSB, Large COSEG, ShapeNetCore, and Human Body. The qualitative results are shown in Table VIII. Our visual representations and data revealed several key observations:

- Even when trained on a small, fully labeled 3D shape dataset, the auxiliary model is still capable of accurately

TABLE VIII
ABLATION STUDY EXPERIMENTS ON NETWORK FRAMEWORK.

Module	Auxiliary Only	No Self-Refine	With Self-Refine
PSB	83.26%	89.37%	**96.10%**
Large COSEG	83.72%	91.60%	**98.49%**
ShapeNetCore	75.36%	82.66%	**88.60%**
Human Body	79.00%	87.15%	**93.15%**

TABLE IX
ABLATION STUDY EXPERIMENTS ON THE THICKNESS OF SCRIBBLE ANNOTATION.

Shrink Rate	1.0	0.7	0.5	0.2
Human	78.04%	84.23%	89.16%	**92.80%**
Ant	85.29%	89.22%	96.97%	**98.17%**
Chair	86.81%	90.42%	96.07%	**98.78%**
Hand	66.10%	72.54%	78.81%	**82.88%**

predicting the 3D shape segmentation labels based on sparse scribble labels.

- The network with the convolutional self-refine module achieves superior prediction results compared to the network without this module. This finding validates our approach of combining the primary network and the auxiliary network to jointly infer label predictions for 3D shapes in dataset S, leading to improved training performance of the primary network.

2) Sensitivities to Scribble Quality: We explored the sensitivity of our method to the quality of scribble annotations in two key aspects. Firstly, we investigated the influence of scribble thickness on experimental results. The segmentation accuracy of four shape categories in the PSB dataset was presented in Table IX, with varying degrees of shrink applied to manually labeled scribbles. Examples of various shrink rates are illustrated in Figure 12. Our observations indicated that as the shrink rate increased, the accuracy of segmentation predictions decreased, which was consistent with our expectations. Furthermore, we noted that even when only one face was labeled with scribble (shrink rate equals 1), the network achieved an accuracy close to half in its predictions.

Secondly, we investigated the influence of scribble shapes on experimental results. As shown in Table X, we presented the effects of four different scribble shapes on the average segmentation accuracy in the PSB dataset, with examples of these scribble shapes depicted in Figure 13. The results showed that while there was some variation in accuracy across different scribble shapes, the effect was not statistically significant. Instead, the proportion of faces covered by scribbles is the primary determinant of experiment performance, as it determines the amount of ground truth label information the algorithm can use.

3) Input Feature Descriptors: In Figure 14 and Table XI, our ablation experiments indicate that the optimal feature vector setup is 122 dimensions. We have attempted to extract features directly from 3D shapes using other networks, but the results were unsatisfactory. Unfortunately, high-quality training data is still insufficient for us to learn effective features for 3D shape processing. This limitation significantly hinders our learning-based approach from achieving more satisfactory results.

TABLE X
ABLATION STUDY EXPERIMENTS ON THE DIFFERENT SHAPES OF SCRIBBLE ANNOTATION.

Scribble Shape	line	dashes	rectangle	cycle
Accuracy	96.10%	95.17%	**96.42%**	96.27%

TABLE XI
ABLATION STUDY OF DIFFERENT INPUT FEATURE DESCRIPTORS.

Descriptors	SDF	SDF +AGD	SDF +AGD +GC	SDF +AGD +GC +WKS	SDF +AGD +GC +WKS +SIHKS
Human	81.57%	85.79%	88.63%	91.28%	**92.80%**
Ant	92.58%	94.15%	96.88%	97.79%	**98.17%**
Chair	88.72%	91.35%	95.78%	97.50%	**98.78%**
Hand	65.80%	68.47%	73.65%	79.44%	**82.88%**

E. Performance

We implemented the proposed algorithm using Matlab, Python, and C++. Our approach was tested on a PC with an Intel Core i7 CPU, 32GB of RAM, and an NVIDIA GeForce GTX 3090 GPU. Our semi-supervised method consists of two main phases, including the training phase and the testing phase. In the training phase, our algorithm takes about 10 minutes to train a single category shape. In the testing phase, segmenting an unlabeled shape takes about 20 seconds. The entire pipeline for a category shape on the PSB dataset takes approximately 20 minutes.

V. LIMITATIONS AND FUTURE WORK

Although the method proposed in this paper is effective in various 3D shape segmentation tasks, it has some limitations. Firstly, the method heavily relies on various hand-crafted feature descriptors and therefore requires manifold 3D shapes as input. In the future, we aim to extend our method to handle non-manifold 3D shapes. Secondly, similar to other learning-based 3D shape segmentation methods, our neural network is limited to the same category of testing 3D shapes as the training 3D shapes to obtain satisfactory segmentation results. We plan to work on generalizing our method to apply across different categories in the future.

VI. CONCLUSION

We propose a novel semi-supervised framework for 3D shape segmentation that leverages a small, fully labeled dataset containing face-level semantic segmentation labels and a sparsely scribble-labeled dataset. Our framework initially trains an auxiliary network to generate initial face-level segmentation labels for the sparsely scribble-labeled dataset, assisting in training the primary segmentation network. During training, the self-refine module improves the labels of the scribble-labeled dataset used to train the primary network by

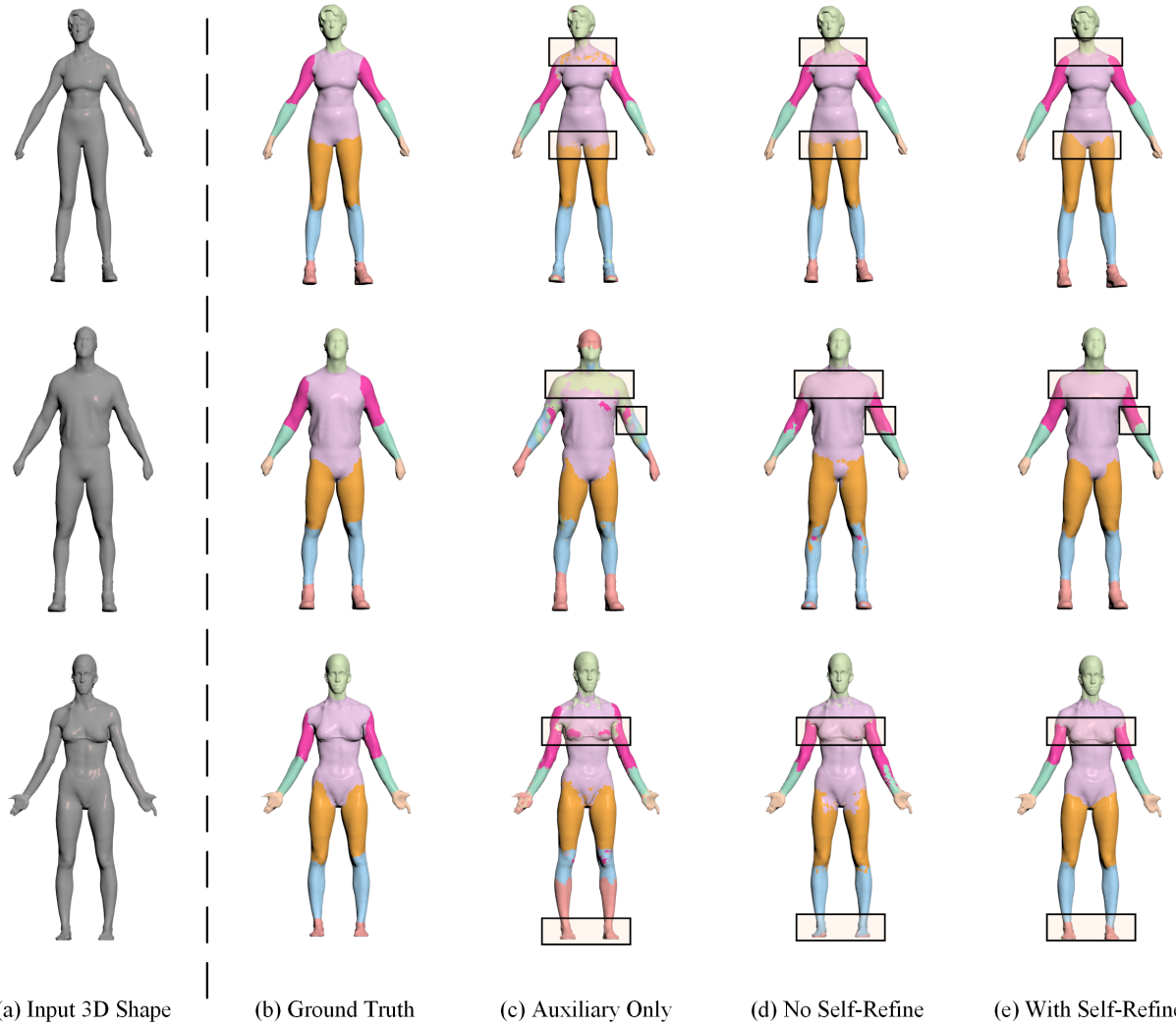


Fig. 11. Ablation Experiments on Network Framework: (a) shows the original unlabeled 3D shape. (b) displays the fully labeled ground truth. (c) presents the prediction results using only the auxiliary network. (d) showcases the prediction results obtained by combining the primary network and the auxiliary network. Finally, (e) illustrates the prediction results with the addition of the convolutional self-refine module to the primary and auxiliary network framework.

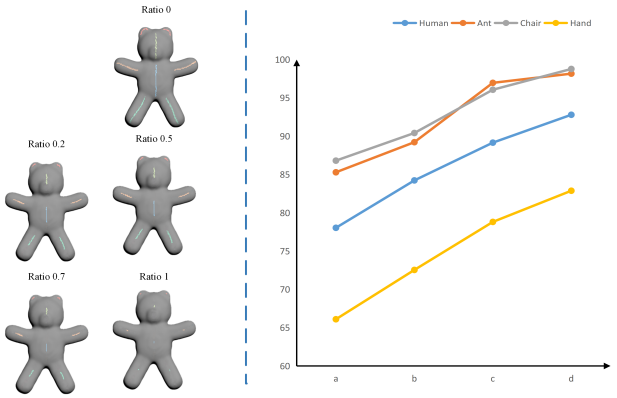


Fig. 12. Ablation Experiments on the thickness of scribble annotation: The figure demonstrates the influence of different scribble shrink rates on the accuracy of segmentation label predictions. The horizontal axis represents different scribble shrink rates: (a) 1.0, (b) 0.7, (c) 0.5, (d) 0.2.

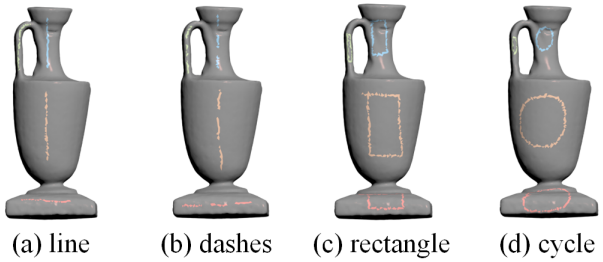


Fig. 13. Ablation Experiments on different scribble annotation shapes: (a) line, (b) dashes, (c) rectangle, (d) cycle.

utilizing the increasingly accurate predictions of the primary model. Our comprehensive benchmark results demonstrate that our proposed method surpasses previous semi-supervised approaches in segmentation performance and achieves comparable performance to fully supervised methods.

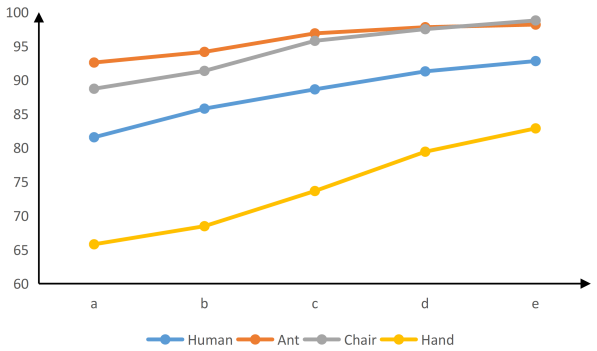


Fig. 14. Ablation Experiments on Feature Descriptors: The figure illustrates the influence of different feature descriptor selections on the accuracy of segmentation label predictions. The horizontal axis represents different combinations of feature descriptors: (a) SDF, (b) SDF + AGD, (c) SDF + AGD + GC, (d) SDF + AGD + GC + WKS, (e) SDF + AGD + GC + WKS + SIHKS.

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] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum, "Mesh editing with poisson-based gradient field manipulation," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 644–651, 2004.

[2] X. Fan, S. Cheng, K. Huyan, M. Hou, R. Liu, and Z. Luo, "Dual neural networks coupling data regression with explicit priors for monocular 3D face reconstruction," *IEEE Transactions on Multimedia*, vol. 23, pp. 1252–1263, 2021.

[3] X. Yang, G. Lin, and L. Zhou, "Single-view 3D mesh reconstruction for seen and unseen categories," *IEEE Transactions on Image Processing*, vol. 32, pp. 3746–3758, 2023.

[4] X. Chen, Y. Guo, B. Zhou, and Q. Zhao, "Deformable model for estimating clothed and naked human shapes from a single image," *The Visual Computer*, vol. 29, no. 11, pp. 1187–1196, 2013.

[5] Y. Yang, W. Xu, X. Guo, K. Zhou, and B. Guo, "Boundary-aware multidomain subspace deformation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 10, pp. 1633–1645, 2013.

[6] A. Mademlis, P. Daras, A. Axenopoulos, D. Tzovaras, and M. G. Strintzis, "Combining topological and geometrical features for global and partial 3-D shape retrieval," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 819–831, 2008.

[7] Z. Kuang, J. Yu, S. Zhu, Z. Li, and J. Fan, "Effective 3-D shape retrieval by integrating traditional descriptors and pointwise convolution," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3164–3177, 2019.

[8] 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.

[9] Z. Wu, Y. Wang, R. Shou, B. Chen, and X. Liu, "Unsupervised co-segmentation of 3D shapes via affinity aggregation spectral clustering," *Computers & Graphics*, vol. 37, no. 6, pp. 628–637, 2013.

[10] M. Meng, J. Xia, J. Luo, and Y. He, "Unsupervised co-segmentation for 3D shapes using iterative multi-label optimization," *Computer-Aided Design*, vol. 45, no. 2, pp. 312–320, 2013.

[11] 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.

[12] 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.

[13] 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.

[14] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3D shape collections," *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 1–12, 2016.

[15] 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.

[16] 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.

[17] S. Cheng, X. Chen, X. He, Z. Liu, and X. Bai, "PRA-Net: Point relation-aware network for 3D point cloud analysis," *IEEE Transactions on Image Processing*, vol. 30, pp. 4436–4448, 2021.

[18] 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.

[19] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 807–832, 2002.

[20] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Symposium on Geometry Processing*, vol. 6, 2003, pp. 156–164.

[21] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[22] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[23] S. Pettie and V. Ramachandran, "An optimal minimum spanning tree algorithm," *Journal of the ACM*, vol. 49, no. 1, pp. 16–34, 2002.

[24] Y. K. Lai, S. M. Hu, R. R. Martin, and P. L. Rosin, "Fast mesh segmentation using random walks," in *Proceedings of ACM Symposium on Solid and Physical Modeling*, 2008, pp. 183–191.

[25] 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.

[26] 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.

[27] C. Lin, L. Liu, C. Li, L. Kobbelt, B. Wang, S. Xin, and W. Wang, "SEG-MAT: 3D shape segmentation using medial axis transform," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 6, pp. 2430–2444, 2020.

[28] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Computational Visual Media*, vol. 7, pp. 187–199, 2021.

[29] F. Yu, K. Liu, Y. Zhang, C. Zhu, and K. Xu, "PartNet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9491–9500.

[30] 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.

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

[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[33] F. Milano, A. Loquercio, A. Rosinol, D. Scaramuzza, and L. Carlone, "Primal-dual mesh convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 952–963, 2020.

[34] D. Smirnov and J. Solomon, "HodgeNet: Learning spectral geometry on triangle meshes," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–11, 2021.

[35] 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.

[36] H. Xu, M. Dong, and Z. Zhong, "Directionally convolutional networks for 3D shape segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2698–2707.

[37] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao, "Meshnet: Mesh neural network for 3D shape representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8279–8286.

[38] X. Li, R. Li, L. Zhu, C.-W. Fu, and P.-A. Heng, "DNF-Net: A deep normal filtering network for mesh denoising," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 10, pp. 4060–4072, 2020.

[39] S.-M. Hu, Z.-N. Liu, M.-H. Guo, J.-X. Cai, J. Huang, T.-J. Mu, and R. R. Martin, "Subdivision-based mesh convolution networks," *ACM Transactions on Graphics*, vol. 41, no. 3, pp. 1–16, 2022.

[40] 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.

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

[42] M. Rong, H. Cui, and S. Shen, "Efficient 3D scene semantic segmentation via active learning on rendered 2D images," *IEEE Transactions on Image Processing*, vol. 32, pp. 3521–3535, 2023.

[43] S. He, X. Jiang, W. Jiang, and H. Ding, "Prototype adaption and projection for few-and zero-shot 3D point cloud semantic segmentation," *IEEE Transactions on Image Processing*, 2023.

[44] Z. Shu, X. Shen, S. Xin, Q. Chang, J. Feng, L. Kavan, and L. Liu, "Scribble based 3D shape segmentation via weakly-supervised learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 8, pp. 2671–2682, 2020.

[45] Y. Zhuang, M. Zou, N. Carr, and T. Ju, "Anisotropic geodesics for live-wire mesh segmentation," *Computer Graphics Forum*, vol. 33, no. 7, pp. 111–120, 2014.

[46] Y. Zhuang, H. Dou, N. Carr, and T. Ju, "Feature-aligned segmentation using correlation clustering," *Computational Visual Media*, vol. 3, no. 2, pp. 147–160, 2017.

[47] A. Tao, Y. Duan, Y. Wei, J. Lu, and J. Zhou, "SegGroup: Seg-level supervision for 3D instance and semantic segmentation," *IEEE Transactions on Image Processing*, vol. 31, pp. 4952–4965, 2022.

[48] Z. Shu, S. Yang, H. Wu, S. Xin, C. Pang, L. Kavan, and L. Liu, "3D shape segmentation using soft density peak clustering and semi-supervised learning," *Computer-Aided Design*, vol. 145, p. 103181, 2022.

[49] 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.

[50] 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.

[51] 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.

[52] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1704–1711.

[53] D. Raviv, M. M. Bronstein, A. M. Bronstein, and R. Kimmel, "Volumetric heat kernel signatures," in *Proceedings of the ACM Workshop on 3D Object Retrieval*. ACM, 2010, pp. 39–44.

[54] 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.

[55] 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.

[56] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "ShapeNet: An information-rich 3D model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[57] O. v. Kaick, H. Zhang, and G. Hamarneh, "Shape segmentation by approximate convexity analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 4, pp. 669–685, 2010.

[58] 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.

[59] C. Lin, L. Liu, C. Li, L. P. Kobbelt, B. Wang, S. Xin, and W. Wang, "SEG-MAT: 3D shape segmentation using medial axis transform," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, pp. 2430–2444, 2022.

[60] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-BERT: Pre-training 3D point cloud transformers with masked point modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 313–19 322.

[61] A. Poulenard and M. Ovsjanikov, "Multi-directional geodesic neural networks via equivariant convolution," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 1–14, 2018.

[62] Y. Yang, S. Liu, H. Pan, Y. Liu, and X. Tong, "PFCNN: Convolutional neural networks on 3D surfaces using parallel frames," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2020, pp. 13 578–13 587.

[63] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov, "DiffusionNet: Discretization agnostic learning on surfaces," *ACM Transactions on Graphics*, vol. 41, no. 3, pp. 1–16, 2022.

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 image processing, computer graphics, digital geometry processing, and machine learning. He has published over 40 papers in international conferences or journals.

Teng Wu is a graduate student of the College of Computer Science and Technology at Zhejiang University. His research interests include image processing, computer graphics, and machine learning.

Jiajun Shen is a graduate student of the School of Software Technology at Zhejiang University. His research interests include computer graphics, geometric processing and computer vision.

Shiqing Xin is an associate 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 image processing, computer graphics, computational geometry, and 3D printing.

Ligang Liu received the BSc and PhD degrees from Zhejiang University, China, in 1996 and 2001, respectively. He is a professor at the University of Science and Technology of China. Between 2001 and 2004, he was at Microsoft Research Asia. Then he was at Zhejiang University during 2004 and 2012. He paid an academic visit to Harvard University during 2009 and 2011. His research interests include geometric processing and image processing. He serves as the associated editors for journals of IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Graphics and Applications, Computer Graphics Forum, Computer Aided Geometric Design, and The Visual Computer. His research works could be found at his research website: http://staff.ustc.edu.cn/lgliu