Hierarchical Self-Supervised Learning for 3D Tooth Segmentation in Intra-Oral Mesh Scans

Zuozhu Liu, *Member, IEEE*, Xiaoxuan He, Hualiang Wang, Huimin Xiong, Yan Zhang, Gaoang Wang, *Member, IEEE*, Jin Hao, Yang Feng, Fudong Zhu, and Haoji Hu, *Member, IEEE*

***Abstract*— Accurately delineating individual teeth and the gingiva in the three-dimension (3D) intraoral scanned (IOS) mesh data plays a pivotal role in many digital dental applications, e.g., orthodontics. Recent research shows that deep learning based methods can achieve promising results for 3D tooth segmentation, however, most of them rely on high-quality labeled dataset which is usually of small scales as annotating IOS meshes requires intensive human efforts. In this paper, we propose a novel self-supervised learning framework, named STSNet, to boost the performance of 3D tooth segmentation leveraging on large-scale unlabeled IOS data. The framework follows two-stage training, i.e., pre-training and fine-tuning. In pre-training, three hierarchical-level, i.e., point-level, region-level, cross-level, contrastive losses are proposed for unsupervised representation learning on a set of predefined matched points from different augmented views. The pretrained segmentation backbone is further fine-tuned in a supervised manner with a small number of labeled IOS meshes. With the same amount of annotated samples, our method can achieve an mIoU of 89.88%, significantly outperforming the supervised counterparts. The performance gain becomes more remarkable when only a small amount of labeled samples are available. Furthermore, STSNet can achieve better performance with only 40% of the annotated samples as compared to the fully supervised baselines. To the best of our knowledge, we present the first attempt of unsupervised pre-training for 3D tooth segmentation, demonstrating its strong potential in reducing human efforts for annotation and verification.**

***Index Terms*— Mesh segmentation, point cloud analysis, self-supervised learning, tooth segmentation.**

I. Introduction

With the development of Computer-Aided Design (CAD) techniques, digital dentistry has attracted tremendous attention with various significant breakthroughs [1], [2], [3], [4], [5], [6]. In digital dentistry, the intraoral scanners (IOSs) are widely used as they can generate a digital impression of the tooth's anatomy by projecting a light source on the dental arches, which are considered more accurate and safer than plaster models. In many dental applications, a fundamental and preliminary step is to precisely segment each individual tooth and the gingiva from 3-Dimensional(3D) dental IOS surfaces [7], which are going to be used for many diagnosis and treatment planning scenarios, such as tooth movement simulation or tooth arrangement planning in orthodontics. Concretely, given an IOS mesh consisting of triangulated faces, 3D tooth segmentation aims to classify each face into different teeth and the gingiva following the FDI standard [8]. However, a single IOS mesh for the upper or low jaw usually consists of more than 100,000 triangular faces. It usually takes about 15 to 30 minutes for an experienced expert to manually or interactively annotate a half jaw, which is undoubtedly cumbersome and labor-intensive [9]. To enable more efficient treatment planning, automated strategies are highly demanded for real-world clinical applications.

Automatic and accurate 3D tooth segmentation remains a challenging task. First, the dentition and tooth appearance vary significantly across patients, e.g., the dental arch shapes (O-, V-, U-shape); tooth numbers (third-molars, hyperdontia), tooth shapes (attrition, macrodontia, crowding teeth) etc. These heterogeneous variations impose significant challenges for achieving robust and accurate performance. Second, the segmentation system needs to generate fine-grained segmentation for high-resolution meshes with over 100,000 faces, while slight mistakes in the tooth-tooth or tooth-gingiva boundaries or failing to recognize tiny tooth parts such as erupted teeth

Manuscript received 13 August 2022; revised 7 October 2022; accepted 5 November 2022. Date of publication 15 November 2022; date of current version 2 February 2023. This work was supported in part by the National Natural Science Foundation of China under Grant U21B2004 and Grant 62106222 and in part by the Zhejiang Provincial Key Research and Development Program of China under Grant 2021C01119. *(Zuozhu Liu and Xiaoxuan He contributed equally to this work.) (Corresponding author: Haoji Hu.)*

Zuozhu Liu is with the College of Information Science and Electrical Engineering, the Stomatology Hospital, School of Stomatology, and the ZJU- UIUC Institute, ZJU-Angelaling Research and Development Institute for Intelligence Healthcare, Zhejiang University, Hangzhou, Zhejiang 310058, China (e-mail: zuozhuliu@intl.zju.edu.cn).

Xiaoxuan He, Hualiang Wang, and Haoji Hu are with the College of Information Science and Electrical Engineering, Zhejiang University, Hangzhou, Zhejiang 310058, China (e-mail: xiaoxuan_he@zju.edu.cn; hualiang_wang@zju.edu.cn; haoji_hu@zju.edu.cn).

Huimin Xiong and Gaoang Wang are with the ZJU-UIUC Institute, and the ZJU-Angelaling Research and Development Institute for Intelligence Healthcare, Zhejiang University, Hangzhou, Zhejiang 314400, China (e-mail: huiminx@zju.edu.cn; gaoangwang@intl.zju.edu.cn).

Yan Zhang is with the School of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: eleyanz@nus.edu.sg).

Jin Hao is with the Harvard School of Dental Medicine, Harvard University, Boston, MA 02138 USA (e-mail: jin_hao@g.harvard.edu).

Yang Feng is with the Angelalign Research Institute, AngelAlign Inc., Shanghai 200011, China (e-mail: fengyang@angelalign.com).

Fudong Zhu is with the School of Stomatology, Stomatology Hospital, Zhejiang University School of Medicine, Hangzhou, Zhejiang 310058, China (e-mail: zfd@zju.edu.cn).

Digital Object Identifier 10.1109/TMI.2022.3222388

might lead to severe issues in subsequent diagnosis. Last but not least, achieving robust and accurate performance across diverse IOS scans usually require large-scale annotated dataset, while a public dataset is not yet available due to privacy issues and the time-consuming annotation process.

Several deep learning based methods are proposed for end-to-end 3D tooth segmentation [2], [9], [10], [11], [12], [13]. Some of them employ hierarchical frameworks to delineate the gingiva and different teeth step by step based on conventional 2D/3D CNNs, which usually suffers from inferior performance. Recent research designs novel architecture for 3D tooth segmentation, however, most of these methods are only trained or validated with a small dataset [13] [12], i.e., less than 50 IOS meshes, as annotating 3D IOS dental surfaces requires complex pipelines and intensive human efforts. Meanwhile, some of them are only applicable to regular IOS scans, such as scans without third-molar, limiting their applications in real-world scenarios. Moreover, when these methods are evaluated in clinical settings, their performance always degrades due to the inferior generalization ability across diverse anatomical tooth features [9].

The aforementioned challenges and limitations of current work motivate us to propose new methods which can achieve better 3D tooth segmentation performance while requiring minimal labor-intensive annotations. Recent research has witnessed the great success of unsupervised pre-training strategies for various computer vision, such as SimCLR [14], BYOL [15], Moco [16], and natural language processing tasks [17]. As for 3D point clouds, several pioneering works also investigate unsupervised pre-training for 3D point cloud processing via occlusion completion, contrastive learning or spatio-temporal representation learning strategies [18], [19]. There also exist some works that adopt self-supervised learning methods for 2D/3D grid-structure medical image analysis, such as [20], [21], [22], [23], [24], and [25]. However, there are still a significant gap between these methods and our tasks. On the one hand, these methods are usually proposed for grid-structure 2D/3D images with contrastive losses based on instance-level image categories, leading to an embedding space where all instances are well-separated. However, the 3D tooth segmentation dataset only contains two half-jaw categories (mandible and maxillary) on non-Euclidean point clouds, while constructing positive and negative pairs over them would not lead to performance gain on the low-level tooth segmentation task. On the other hand, directly applying existing methods on 2D/3D grid images to the 3D tooth point cloud segmentation task did not lead to significant performance improvement. Domain-specific designs that explore the geometric features of 3D tooth point clouds or the morphological structures in the IOS are required to obtain better performance for 3D tooth segmentation, such as different data augmentation strategies or architecture designs.

In this paper, we propose a novel method to boost the performance of 3D tooth segmentation with an unsupervised pre-training strategy that leverages large-scale unlabeled IOS meshes. The method is termed STSNet, i.e., Self-supervised Tooth Segmentation Network. To cope with the high-resolution mesh data, we formulate the segmentation task over 3D dental meshes as a fine-grained point cloud semantic segmentation task as in [12] and [9], avoiding approximation errors in voxel-based methods. We design a hierarchical self-supervised learning framework with three contrastive losses, i.e., the point-level, region-level and cross-level contrastive losses. These losses perform contrastive learning on different scales, i.e., point-level contrastive loss for local representation learning of individual points, region-level loss to capture global region contextual and morphological features with an additional Relation-Shape CNN network [26], and cross-level loss to further bridge the global-local gap by allowing region representations to guide representation learning of individual points. By doing so we encourage each point to not only keep point-level consistencies but also explicitly maintain contextual consistencies, which are important for accurate tooth segmentation. With the Dynamic Graph CNN (DGCNN) [27] as backbone, the STSNet is first pre-trained over a set of predefined matched points from different augmented views obtained via a new augmentation strategy. Afterwards, the pre-trained network is slightly modified to adapt to the downstream semantic segmentation task and fine-tuned on a small labeled dataset for inference.

To evaluate the effectiveness of our method, we construct a large 3D IOS mesh dataset, consisting of 12,000 unlabeled IOS meshes and 1,000 labeled meshes. Extensive experiments reveal that STSNet can achieve an mIoU of 89.88%, which can be further improved to 93.12% with a widely-used graph-cut post-processing strategy, significantly outperforming all supervised counterparts when trained with the same amount of labeled samples. The performance gain becomes more remarkable when only a tiny amount of labeled samples are available, e.g., STSNet outperforms the training-from-scratch supervised counterpart by a large margin of 21.93% mIoU when only 1% of labeled samples are available. Furthermore, STSNet achieves segmentation performance better than the best fully supervised baselines with only 40% of the annotated samples. This work is an extension to our preliminary conference paper [28] but with substantial novel components. The hierarchical constrastive learning framework is significantly different from the naive self-supervised PointInfoNCE loss in [19] and [28]. The detailed strategy to generate augmented views is improved as well. The unlabeled dataset is enlarged with more comprehensive results. To the best of our knowledge, our work, including [28], is the first attempt to employ unsupervised pre-training methods for 3D tooth segmentation, exhibiting strong potential to reduce human effort for annotation and verification.

The rest of the paper is organized as follows. We introduce the related work on self-supervised learning and 3D shape segmentation in Section II, while previous work regarding tooth segmentation is already discussed in this section. The details about the proposed self-supervised learning framework is presented in Section III. In Section III, we list the dataset, experimental setup and implementation, and experiment results with visualizations and discussions. We conclude the paper in Section VI.

II. Related Work

A. Self-Supervised Learning

Self-supervised learning has emerged as a common paradigm to learn powerful representations from unlabeled data, where the supervisory signals can be generated based on the structure of the data itself [29], [30], [31]. The representation pre-trained through self-supervision could be utilized for fine-tuning multiple downstream supervised tasks with better generalization and calibration [32], [33]. Contrastive learning is a typical self-supervised learning approach that has recently achieved great success and become a milestone in the field of visual representation learning [14], [16], [34]. The idea of these models is to pull representations from different augmented views of the same sample closer, while pushing representations from other samples apart. Recent work transfers this learning strategy to texts and graphs with different network architectures and augmenting methods for self-supervised representation learning. [35], [36]. There are also several pioneering works that explore self-supervised learning for 2D/3D grid-structured medical data [20], [21], [22], [23], [24], [25]. As for 3D grid-structured medical image segmentation, most of these methods group the images of each volume into S partitions, each containing consecutive images [22], or split each volume into slices [23], [24], [25], which are subsequently converted to image-level contrastive learning with 2D contrastive learning strategy while adding some extra domain-specific losses, e.g., losses based on the temporal feature of the volumetric data. The current unsupervised methods for 3D medical images usually convert them, such as CT or MRI, into 2D images, and then add the domain-specific characteristics from each modality. However, these method is not applicable for 3D tooth segmentation which is defined over high-resolution non-Euclidean point clouds with complicated anatomical structures. In this work, we present the first work to systematically explore whether and how contrastive learning could be leveraged to solve the 3D tooth segmentation task in a self-supervised manner.

B. 3D Shape Segmentation

There is a substantial amount of work for 3D shape segmentation. Some work first transforms the 3D surfaces or point clouds into voxels [37], which are subsequently processed by 3D convolutional neural networks. Though there exist efficient 3D CNN implementations, the computational overhead is still large, and as mentioned above, some additional approximation errors are introduced during voxelization, leading to inferior performance for tasks requiring fine-grained results on very high-resolution data such as tooth segmentation. Another line of work takes the raw surface/mesh/point cloud as input to train a deep neural network, e.g., PointNet [38], PointNet++ [39], DGCNN [27], RPNet [40], CurveNet [41]. For example, PointNet designs a novel type of neural network that directly operates on point clouds while preserving permutation invariance of points. PointNet++ further introduces a hierarchical network based on PointNet on a nested partitioning of the input points. DGCNN proposes a novel EdgeConv block to learn semantic representation with proximities over different hierarchies. Recently, transformer based architectures are also proposed for 3D point cloud processing [42]. Though promising, their performance is not satisfactory when directly adapted to 3D tooth segmentation. Moreover, there are some work exploring unsupervised pre-training in 3D vision [19], [43], [44]. However, they are proposed for general 3D point clouds especially with experimental results on standard shape recognition dataset such as ModelNet40 [45] or ShapeNetPart [46], while 3D IOS surfaces differ from such nature objects significantly in terms of resolution, shape and structures. Second, the tooth segmentation task requires fine-grained results on high-resolution data, while previous work is usually examined with coarse-level predictions.

C. 3D Tooth Segmentation

A lot of works have launched attempts to address the 3D tooth segmentation task in IOS meshes. Traditional geometry-based methods extract hand-crafted features such as curvatures from IOS meshes to design decision rules for segmentation [47], [48], [49], [50], [51]. However, these methods are not fully-automatic and usually require human intervention for interactive segmentation or post processing to correct the inferior results. Recently, many pioneering deep learning based methods [10] have been proposed with superior performance for 3D tooth segmentation. Some works first extract predefined features and subsequently apply the 2D or 3D convolutional neural networks for 3D tooth semantic segmentation [11]. Though with better performance than traditional geometry-based methods, the hierarchical segmentation procedures in these works are usually time-consuming and error-prone, as mistakes occurring in former steps could not be corrected in later steps. The performance is further boosted with methods which design specific neural network architectures for end-to-end tooth segmentation, such as MeshSegNet [12], DC-Net [9], TSegNet [13], Mask-MCNet [2] etc. In particular, MeshSegNet integrates graph-constrained learning modules to extract multi-scale local contextual features; DC-Net presents an accurate, efficient, and fully automated deep learning model with uncertainty estimations; TSegNet segments the tooth based on a two-stage network; Mask-MCNet design a network based on the Mask-RCNN for tooth instance segmentation. However, these methods have limitations in terms of the dataset size and generalization ability across diverse IOS scans as presented in Section I. Though the DC-Net presents attempts towards clinically applicable solutions, it relies on a large-scale annotated dataset which is time-consuming to collect and not publicly available due to privacy issues [9]. As a result, it's hard for the community to make further advancements to meet the requirements for clinical usages. In our method, we propose a novel self-supervised learning framework that can take advantage of large-scale unlabeled dataset to achieve better performance than the supervised counterparts.

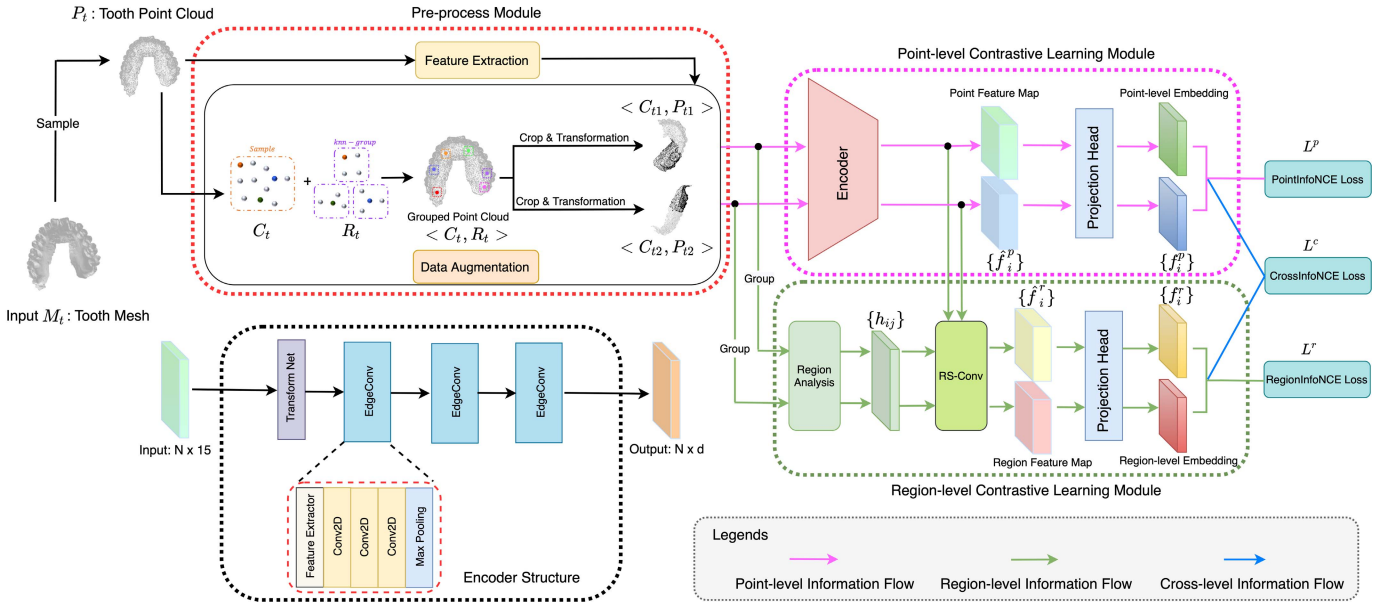

Fig. 1. The proposed framework for unsupervised pre-training. First, we sample a point cloud P_t from a mesh M_t. Second, a pre-process module, including the feature extraction and augmented data processing, is devised to generate two input pairs in the region level (C_{t1}, C_{t2}) and point level (P_{t1}, P_{t2}). Afterwards, the two pairs are fed into the region- and point-level contrastive learning modules to obtain the RegionInfoNCE and PointInfoNCE loss values, respectively. Finally, the CrossInfoNCE loss value is calculated by two cross pairs (P_{t1}, C_{t2}) and (P_{t2}, C_{t1}).

III. Method

A. Overview

Given a 3D IOS mesh composed of many triangulated faces, 3D tooth segmentation aims to classify each face into different teeth and the gingiva following the FDI standard. Mathematically, for each face f_i in the mesh, we want to annotate it with a label y_i, where $y_i \in \{0, 11-18, 21-28, 31-38, 41-48\}$ denotes the gingiva and FDI notations for the 32 permanent teeth, respectively. Our method includes two steps: unsupervised pre-training and supervised fine-tuning. In unsupervised pre-training, we first generate two augmented views of each unlabeled 3D IOS mesh and feed them into the segmentation backbone. Three different levels of contrastive losses are designed for unsupervised representation learning on a set of predefined matched points. Afterwards, the pre-trained encoder is further fine-tuned in a supervised manner with a small number of labeled 3D IOS meshes.

B. Feature Extraction and Augmented Data Processing

1) Feature Extraction: Let $\mathcal{X} = \{M_t\}_{t=1}^{L}$ be the dataset with L 3D IOS meshes where $M_t = (V, F)$ denotes the t-th sample with V and F as mesh vertices and faces, respectively. We sample a point cloud $P_t \in \mathbb{R}^{N\times 15}$ from M_t with N points each associated 15-dimensional features. The 3D coordinate of each point is the center of the corresponding face, which is denoted as $h_c = [x_0, y_0, z_0] \in \mathbb{R}^3$. We further extract more geometrical features from the original mesh for each point. In particular, we compute the normal vector $h_n \in \mathbb{R}^3$, and a face shape descriptor $h_s \in \mathbb{R}^9$ as suggested in [9]. For faces with three vertices $v_i = [x_i, y_i, z_i]_{i=1}^3$ and a face center $h_c = [x_0, y_0, z_0]$, the face shape descriptor is simply defined as $h_s = \text{Concat}([v_i - h_c]_{i=1}^3)$, where Concat() represents concatenate operation for vectors. Finally, we concatenate the three features together as the feature vector of each point in our point cloud, leading to a 15-dimensional feature vector $h = \text{Concat}(h_c, h_n, h_s) \in \mathbb{R}^{15}$.

2) Augmented Data Preprocessing: It is not uncommon to generate asymmetric augmented input pairs for better representation learning in self-supervised pre-training with contrastive learning strategies. As for 3D meshes or point clouds, the augmented input pair, which usually contains two augmentations with different views of the same input, should bring much more abundant and diverse training examples while discouraging the model from learning simple equivariance of the geometric transformation. Consequently, we generate two different views as our pre-training input. The pipeline to generate asymmetric input pairs is elaborated as follows.

We define two input pairs in the region level and point level. For the input pairs in the region level, we first sample G center points from the holistic point cloud via farthest point sampling (FPS). The k-nearest neighbor (kNN) algorithm is then used to select the n nearest neighbor points for each center point, grouping G regions, denoted as $R_t \in \mathbb{R}^{G\times(n+1)\times 3}$. The high-dimensional R_t is represented as low-dimensional center points set $C_t \in \mathbb{R}^{G\times 3}$, of which one center point indicates its corresponding point region. We then randomly crop G_0 regions twice from the above G regions, denoted as $R_{t1} \in \mathbb{R}^{G_0\times(n+1)\times 3}$, $C_{t1} \in \mathbb{R}^{G_0\times 3}$ and $R_{t2} \in \mathbb{R}^{G_0\times(n+1)\times 3}$, $C_{t2} \in \mathbb{R}^{G_0\times 3}$, respectively. C_{t1} and C_{t2} are the input pairs in the region level. The correspondence mapping between regions from C_{t1} and C_{t2} are computed as $P^{mr} = \{(i, j)\} = \Phi_r(C_{t1}, C_{t2})$, where i and j are the index of the matched region center points $x_i^r \in \mathbb{R}^3$ in C_{t1} and $y_j^r \in \mathbb{R}^3$ in C_{t2}, respectively.

For the input pairs in the point level, we first uniformly downsample R_{t1} and R_{t2} to with N_0 points, as they may contain different numbers of points because of the overlap

among different regions during the above region generation process. Then we extract the corresponding features to obtain two asymmetric augmented input point clouds $P_{t1} \in \mathbb{R}^{N_0 \times 15}$ and $P_{t2} \in \mathbb{R}^{N_0 \times 15}$. The correspondence mapping between points from the two point clouds are computed as $P^{mp} = \{(i, j)\} = \Phi_p(P_{t1}, P_{t2})$, where i and j are the index of the matched points $x_i^p \in \mathbb{R}^3$ in P_{t1} and $y_j^p \in \mathbb{R}^3$ in P_{t2}, respectively. As for Φ, We follow the method of point cloud registration [52], [53], [54], i.e., we regard x_i^p and y_j^p as corresponding points when $\|x_i^p - y_j^p\|_2$ is less than a certain threshold (empirically set as 0.75 in our experiments).

3) Transformation: We apply different transformations on the 3D point clouds to generate different augmented views. Mathematically, we define the transformation $\mathcal{T} = [R|t|S]$, in which $R \in SO(3)$ (3D rotation group in geometry) denotes the rotation, $t \in \mathbb{R}^3$ denotes translation, and S denotes scaling operations, respectively. For rotation R, we rotate point clouds with random angles (0 to 360°) around an arbitrary axis. Meanwhile, the function t is devised to translate point clouds globally in the coordinates. The random scale function S is designed to scale point clouds with a factor randomly chosen from the range [0.8, 1.2].

C. Unsupervised Pretraining

As shown in Fig. 1, our unsupervised pre-training framework employs a hierarchical contrastive learning architecture, which enables the encoder to learn three level (i.e. point, region, cross) consistent representations by shrinking the distance between samples from the same asymmetric pair in the hidden space. Specifically, our framework includes the point-level and region-level contrastive learning module to learn corresponding representations and three contrastive loss function for unsupervised training.

1) Point-Level Contrastive Learning: We design the point-level encoder to learn feature representations of the extracted point clouds from 3D tooth data, as shown in Fig. 1. The encoder is inspired by the Dynamic Graph CNN (DGCNN) [27] with modifications to adapt to the 3D IOS data, which is of much higher resolution and morphological complexity. Let's consider P_{t_1} only. It is firstly transformed into a standard feature space with the Transform Net [38]. Second, it is fed to three consecutive Edge-Conv blocks [27], which consist of a feature extractor based on kNN strategy, three 2D convolutional layers, and a max-pooling aggregation operation. Based on an explicit local graph among neighborhood points defined by kNN, the Edge-Conv block updates the edge features with convolutional operations. The features used for kNN are the corresponding output from the previous block, leading to updated proximity defined on different hidden representations. Hence, the stacked Edge-Conv blocks can learn local features in the bottom layers and global semantic features in the top layers. With the concatenated representation from different layers, our backbone is able to capture both local topological geometry and global features for every point in P_{t_1}. Such representations are projected to a consistent hidden space with a projection head (i.e., a Multilayer Perceptron) for subsequent contrastive representation learning, following the standard conventions in many contrastive learning paradigms.

The InfoNCE loss [34] is proposed and has been widely used for unsupervised pre-training in 2D vision tasks. It is adopted by contrastive learning frameworks to conduct a dictionary query process. Here we define the PointInfoNCE loss [19], [28] over points in the two augmented point clouds. We define the matched point pairs (i, j) in P^{mp} as positive pairs, whose features $f_i^p \in \mathbb{R}^d$ and $f_j^p \in \mathbb{R}^d$ are obtained via the encoder and projection head. We further define (i, k) as negative pairs for $\forall (i, k) \in P^{mp}, k \neq j$. In this case, we are considering points that have at least one matched point pairs in P^{mp} as the negative samples, ignoring all other non-match points for more efficient loss computation. Given the positive and negative pairs, the contrastive learning loss is defined as:

$$\mathcal{L}^p = -\frac{1}{|P^{mp}|} \sum_{(i,j) \in P^{mp}} \log \frac{exp(f_i^p \cdot f_j^p / \tau)}{\sum_{(\cdot,k) \in P^{mp}} exp(f_i^p \cdot f_k^p / \tau)}, \tag{1}$$

where $f_i^p \cdot f_j^p$ denote the dot product between f_i^p and f_j^p, and τ is the temperature hyperparameter.

2) Region-Level Contrastive Learning: In the previous section, we introduce a method to maintain point-level consistency. However, many regions of the tooth have their own inherent characteristics (e.g., the gingival margin, corona dentis). In order to improve the sensitivity of the encoder to tooth region morphology, we propose a module to enhance consistent region-level representations. To obtain an inductive region representation with explicit reasoning about the spatial layout of points, we adopt the Relation-Shape CNN (RS-Conv) network on top of the output from the point-level encoder [26]. Specifically, given an augmented point cloud $P_{t1} \in \mathbb{R}^{N_0 \times 15}$ and its center points $C_{t1} \in \mathbb{R}^{G_0 \times 3}$, we first group the point cloud based on the coordinates of center points $\{x_i^r\}_{i=1}^{G_0}$ with the kNN strategy. Then, we define the relational coefficient term vector as $h_{ij} = \text{Concat}(\|x_i^r - x_j^p\|_2, x_i^r - x_j^p, x_i^r, x_j^p, h_j^p) \in \mathbb{R}^{13}$, where h_j^p is the normal of neighbor x_j^p. Hence, h_{ij} includes geometric priors about both 3D coordinates and normals between the center point x_i^r and its neighbor x_j^p, which is further processed with RS-Conv layers to learn high-level relations among the points to better encode shape information. The computation is defines as:

$$\hat{f}_i^r = \sigma(\mathcal{A}(\{\mathcal{M}(h_{ij}) \cdot \hat{f}_j^p, \quad \forall x_j^p \in \mathcal{N}(x_i^r)\})), \tag{2}$$

where $\mathcal{N}$ is a group function based on the kNN strategy and $\hat{f}_j^p$ is the feature vector for x_j^p, extracted from the point-level encoder. σ is a non-linear activator. Here $\hat{f}_i^r$ is obtained by first transforming the features of all the neighbor points in $\mathcal{N}(x_i^r)$ with $\mathcal{M}(h_{ij})$, where $\mathcal{M}$ is a shared multi-layer perception (MLP) whose goal is to obtain high-level relations between center points and their neighbor points. Finally, the aggregation function $\mathcal{A}$ achieves the permutation invariance of point set, which is instantiated as summation aggregation and multi-layer perceptrons (MLP). This module can capture a contextual region feature expression from predefined geometric priors as in h_{ij} between point cloud P_{t1} and its center points C_{t_1}. Such high-level features are uniformly

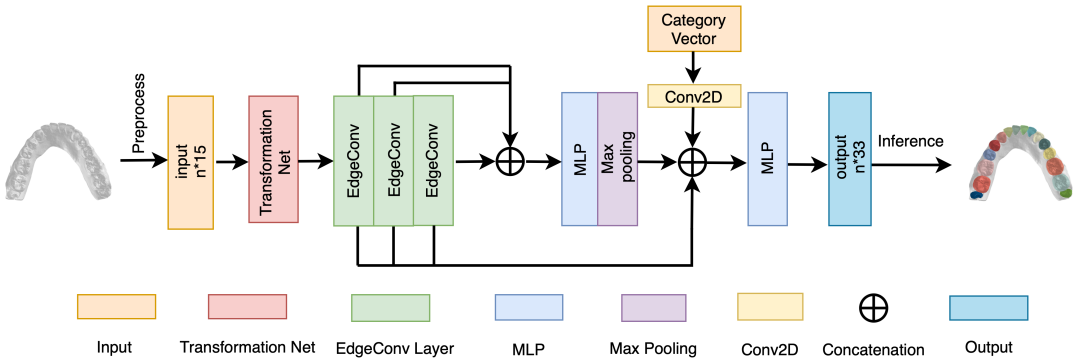

Fig. 2. The architecture of supervised fine-tuning: Input → Transform Net → EdgeConv → EdgeConv → EdgeConv → Conv2D[1024] → maxpool → Conv2D[256] → Dropout → Conv2D[256] → Dropout → Conv2D[128] → Output. The number inside the bracket denotes the number of filters for 2D CNNs, e.g., Conv2D[64] means a convolutional layer with 64 filters. Unless otherwise indicated, all the Conv2D layers use a kernel size of [1,1] and a stride size of [1,1] with batch normalization and Mish activation.

projected to consistent hidden space with a projection head (i.e., a Multilayer Perceptron), obtaining $\{f_i^r\}_{i=1}^{G_0}$, $f_i^r \in \mathbb{R}^d$.

Along the lines of point-level contrastive learning strategy, positive pair for region i is defined as the matched region pair $(i, j) \in P^{mr}$. Negative pairs for region i are presented as non-matched region pairs $\forall(i, k) \in P^{mr}, k \neq j$. In addition, only regions that have at least one negative pair are considering for efficient loss computation. Given all positive and negative pairs, the contrastive learning loss is defined as follows:

$$\mathcal{L}^r = -\frac{1}{|P^{mr}|} \sum_{(i,j)\in P^{mr}} \log \frac{exp(f_i^r \cdot f_j^r/\tau)}{\sum_{(\cdot,k)\in P^{mr}} exp(f_i^r \cdot f_k^r/\tau)}. \tag{3}$$

3) Cross-Level Contrastive Learning: In addition to the above point-level and region-level representation learning module, we further propose a novel dense cross-level contrastive loss. We assume that well-represented points should exhibit the following properties: embedding for points within a region are consistent, while embedding for points from different regions should be sufficiently discriminative. To this end, given the embedding of point f_j^p from P_{t1} and region f_i^r from C_{t2}, we using the following formula to calculate the cross-level contrastive loss:

$$\mathcal{L}_i^c = -\frac{1}{|\mathcal{N}(x_i^r)|} \sum_{x_j^p \in \mathcal{N}(x_i^r)} \log \frac{exp(f_i^r \cdot f_j^p/\tau)}{\sum_{x_k^p \in P_{t1}} exp(f_i^r \cdot f_k^p/\tau)}, \tag{4}$$

where x_i^r is the i-th center point in C_{t2} and $\mathcal{N}(x_i^r)$ is the corresponding region points when transform x_i^r into the coordinate system of P_{t1}. Note that we perform cross-level contrastive learning between P_{t1} and C_{t2} rather than C_{t1}. This is because the representations in P_{t1} and C_{t1} are already highly correlated as they shared the same point-level encoder, while contrastive learning over P_{t1} and C_{t2} can bring additional guidance. Similarly, we define the cross-level contrastive loss for P_{t2} and C_{t1}. Overall, with the guidance from higher-level region embeddings, representations between points and their corresponding region centers are encouraged to be consistent. In contrast, point representations are pulled away from other region centers. The final cross-level contrastive learning loss is defined as:

$$\mathcal{L}^c = \frac{1}{2}\left(\frac{1}{|C_{t1}|}\sum_{i\in C_{t1}} \mathcal{L}_i^c + \frac{1}{|C_{t2}|}\sum_{i\in C_{t2}} \mathcal{L}_i^c\right). \tag{5}$$

4) Loss Function: Optimizing over this InfoNCE loss function would minimize the distance between positive pairs while maximizing the distance between negative pairs, leading to good representations for further tooth semantic segmentation. Now we define the total loss L_{total} based on the above loss functions:

$$\mathcal{L}_{total} = \lambda\mathcal{L}^p + (1-\lambda)\mathcal{L}^r + \beta \cdot \mathcal{L}^c, \tag{6}$$

where λ and β are weight hyper-parameters.

D. Supervised Fine-Tuning

The unsupervised pre-trained backbone is further modified and fine-tuned in a supervised manner for the downstream 3D tooth segmentation task. In particular, we use a one-hot categorical vector to denote the maxillary and mandible for the input half jaws, which is prior knowledge to avoid confusion between them during inference. The one-hot vector is further embedded with a convolutional layer and concatenated with the point-wise representations from the pre-trained backbone. The fused representation is used for semantic segmentation over 32 permanent teeth and the gingiva with a multilayer perceptron composed of two fully-connected layers and a dropout layer with a keep probability of 0.4. We use the cross-entropy loss for supervised fine-tuning. The overall deep learning architecture is illustrated in Fig. 2. For the categorical vector, we first cope it with a convolution layer with 64 filters and then feed the output into our backbone. Based on the above architecture, the network is capable of handling 3D teeth data with much higher resolution and morphological complexity.

During fine-tuning, the pre-trained weights serve as initial weights for the supervised backbone, leading to much faster convergence and better performance as shown in experiments. As for inference, we can not feed all the points in IOS meshes (e.g., 100,000+ points) to our network due to overloaded GPU memory, while performing multi-step inference for each of the

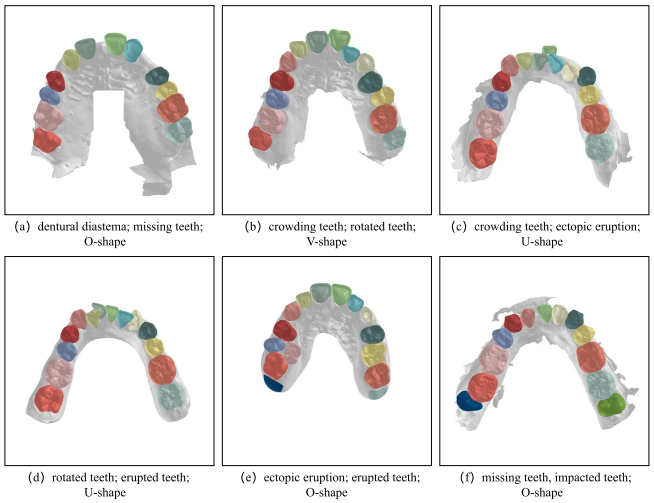

Fig. 3. Visualization of IOS scans with diverse morphological features or diseases.

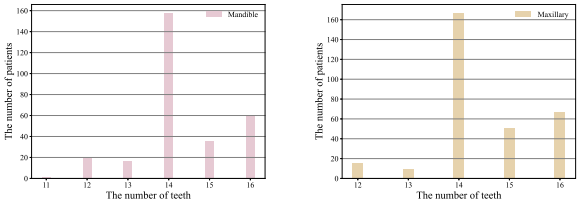

Fig. 4. Statistics of the number of teeth in each IOS on the training set (600 patients).

10,000 points is quite time-consuming as well, e.g., we need 10 inference steps for 100,000 points. In this work, we only inference 40,000 randomly sampled points for each mesh, and use a simple kNN based voting mechanism to generate semantic labels for all the rest points. Such a strategy brings with some performance degardation compared to the multi-step method, but with better efficiency. More investigation of this inference strategy is reported in the experiments.

IV. Experiment

A. Implementation Details

1) Dataset and Experimental Setup: We collect a large 3D IOS tooth mesh dataset, which consists of 12,000 unlabeled and 1,000 labeled 3D IOS mesh data from patients in China. The IOSs are aligned with predefined templates to transform to roughly the same reference positions during pre-processing. Fig. 3 exhibits the diversity of our dataset. The dentition and tooth appearance vary significantly across patients, e.g., the dental arch shapes (O-, V-, U-shape); tooth numbers (missing teeth), tooth shapes (crowding/erupted teeth) etc. The statistics of the number of tooth in each IOS in the training set are displayed in Fig. 4, exhibiting the diverse numbers of tooth across different patients. As for supervised training, We randomly split the labeled data to 60% for training, 20% for validation and 20% for testing in which the number of the mandible and the maxillary jaws are not strictly equal.

In unsupervised pre-training period, we use the SGD optimizer with learning rate $\eta_1 = 0.1$ and an exponentially decay factor 0.99. The hyperparameter $\tau = 0.07$ in three contrastive losses and. During fine-tuning, we use SGD with an initial learning rate $\eta_2 = 0.1$ that decays until 0.001 with cosine annealing. We use $k = 25$ for the kNN step in EdgeConv blocks, and the network is trained with a batch size $bs = 4$ over $N_0 = 8,000$ points and $G = 1,024, G_0 = 717$ regions in pre-training, and $N = 20,000$ points in fine-tuning. We set $\lambda = 0.5$ and $\beta = 0.2$. The network is pretrained with 120 epochs and finetuned with 400 epochs.

2) Post-Processing: The segmentation results produced by deep neural networks may be coarse around the tooth-tooth and teeth-gingival boundaries. Meanwhile, some isolated false predictions also occur. Hence, it is common to adopt the graph-cut based post-processing strategy which could significantly refine the segmentation [9], [10], [12]. We report the results with and without graph-cut smoothing in our experiments.

3) Metrics: We comprehensively evaluate the performance of our method with various metrics, i.e., mIoU, Dice Similarity Coefficient (DSC), and point-level classification accuracy, where a higher value indicates better segmentation performance. All of these metrics are computed following the conventional definitions.

B. 3D Tooth Segmentation Performance

We compare our method with extensive baselines in recent works (e.g., PointNet [38], PointNet++ [39], DGCNN [27], MeshSegNet [12] and DC-Net [9]). The results are reported in Tab. I. We use $*$ to denote methods with the graph-cut smoothing and † to denote methods with large unlabeled dataset (12,000 unlabeled mesh). Our STSNet† achieves state-of-the-art performance compared with all supervised baselines which are trained with the same amount of annotated samples. In particular, the STSNet† also surpasses the modified DGCNN with a significant improvement of 3.33% mIoU and 2.84% DSC. This demonstrates that our unsupervised pre-training method can achieve non-trivial performance improvement, even with the same backbone as its supervised counterparts. Another remarkable achievement is that the STSNet† can achieve better performance than the DC-Net*, which already integrated a graph-cut smoothing, with 1.18% DSC and 0.37% mIoU. STSNet†* further yields 93.12% mIoU and 94.85% DSC with graph-cut smoothing, mainly attributing to the great improvement over the mandible segmentation. Such a performance is extraordinarily better than all supervised baselines.

Moreover, we compare our method with several classical self-supervised methods (e.g., SimCLR [14], BYOL [15], Moco [16]). All above methods are trained with the same data augmentation strategy and backbone as our STSNet. We follow the same settings with Moco and BYOL to transfer our pre-trained encoder to the downstream task (e.g., segmentation). The results are in Tab. I. The best-performing one (SimCLR) can only achieve 1.16% mIoU improvement over the corresponding supervised counterpart (DGCNN), which is still inferior than the best supervised

TABLE I

SEGMENTATION RESULTS OF STSNET AND BASELINES. ∗ DENOTES METHODS WITH GRAPH-CUT SMOOTHING; ◇ DENOTES METHODS WITH DOMAIN-SPECIFIC DATA AUGMENTATION IN OUR DATASET; † DENOTES PRE-TRAINING WITH LARGER DATASET(12,000 UNLABELED DATA)

Method	Mandible			Maxillary			All		
	Acc	mIoU	DSC	Acc	mIoU	DSC	Acc	mIoU	DSC
Supervised methods									
PointNet	87.52	68.23	76.09	91.26	75.18	81.56	89.65	72.19	79.21
PointNet++	88.16	69.35	76.32	91.86	77.62	83.26	90.27	74.06	80.28
MeshSegNet	89.41	71.34	77.99	91.11	77.69	83.68	90.38	74.96	81.23
DGCNN	95.25	84.31	88.18	96.25	88.25	91.37	95.82	86.55	90.00
DC-Net	95.32	84.81	88.52	96.40	89.26	92.28	95.94	87.34	90.67
DC-Net*	92.74	85.44	88.07	96.74	92.58	94.37	95.02	89.51	91.66
Unsupervised methods									
SimCLR◇	95.68	85.62	89.27	96.56	89.29	92.28	96.18	87.71	90.99
Moco◇	95.58	84.84	88.40	96.43	88.37	91.44	96.07	86.85	90.13
BYOL◇	94.78	84.48	88.37	96.03	88.71	91.84	95.49	86.89	90.35
STSNet	**96.35**	**88.58**	**91.80**	**96.81**	**90.67**	**93.42**	**96.61**	**89.77**	**92.72**
STSNet†	**95.90**	**88.42**	**91.72**	**96.89**	**90.98**	**93.69**	**96.47**	**89.88**	**92.84**
STSNet†*	**96.58**	**91.54**	**93.49**	**97.71**	**94.31**	**95.88**	**97.22**	**93.12**	**94.85**

TABLE II

SEGMENTATION PERFORMANCE WITH LIMITED LABELED TRAINING DATA FOR FINE-TUNING

Data Ratio	Train Strategy	Mandible			Maxillary			All		
		Acc	mIoU	DSC	Acc	mIoU	DSC	Acc	mIoU	DSC
1%	from scratch	61.58	36.81	45.81	73.18	51.04	60.20	68.19	44.92	54.01
	Our†	86.95	61.91	70.14	89.32	70.57	76.97	88.30	66.85	74.03
5%	from scratch	87.59	66.31	72.79	89.92	72.49	78.15	88.92	69.83	75.84
	Our†	90.12	69.11	75.27	91.52	76.39	81.38	90.92	73.26	78.75
10%	from scratch	89.76	71.92	77.53	91.66	76.12	81.15	90.84	74.32	79.59
	Our†	92.68	77.34	82.04	93.76	80.98	85.31	93.30	79.41	83.90
20%	from scratch	90.77	72.97	78.18	93.43	79.62	83.92	92.29	76.76	81.45
	Our†	93.57	79.62	84.58	94.66	84.59	88.69	94.19	82.45	86.92
40%	from scratch	93.54	79.73	84.01	95.53	86.16	89.56	94.68	83.40	87.17
	Our†	96.00	86.23	89.57	96.78	89.81	92.42	96.45	88.27	91.19
100%	from scratch	95.25	84.31	88.18	96.25	88.25	91.37	95.82	86.55	90.00
	Our†	**95.90**	**88.42**	**91.72**	**96.89**	**90.98**	**93.69**	**96.47**	**89.88**	**92.84**

TABLE III

SEGMENTATION PERFORMANCE OF STSNET PRE-TRAINED WITH DIFFERENT AMOUNTS OF DATA

Data Ratio	All		
	Acc	mIoU	DSC
10%	96.44(+0.62)	89.40(+2.85)	92.38(+2.38)
50%	96.61(+0.79)	89.77(+3.22)	92.72(+2.72)
100%	96.47(+0.65)	89.88(+3.33)	92.84(+2.84)

DC-Net model. In contrast, STSNet achieves 3.22% mIoU improvement, which already surpasses the best supervised model, demonstrating its great effectiveness on 3D tooth segmentation.

We further investigate the effect of pre-training with different amounts of unlabeled data, with results in Tab. III. When using 10% of unlabeled 3D IOS data (1,200 samples) during pre-training, STSNet still achieves 96.44% accuracy, 89.40% mIoU and 92.38% DSC. Compared to the supervised DGCNN model, it has 2.85% mIoU improvement, revealing that STSNet can still achieve impressive performance even using a limited number of unlabeled samples. Meanwhile, with the increasing amount of unlabeled pre-training data, all the evaluation metrics are constantly improved, convincingly demonstrating the effectiveness of our method.

We also conduct experiments to evaluate the effectiveness of using different amounts, i.e., 1%, 5%, 10%, 20%, 40%, 100%, of the labeled data during fine-tuning, with results shown in Tab. II. We use 'from scratch' to denote the our DGCNN backbone trained without weight initialization from the pretrained STSNet. When trained with only 1% labeled data, DGCNN trained from scratch can only achieve 68.19% accuracy, 44.92% mIoU, and 54.01% DSC. In stark contrast, our method significantly outperforms it with 88.30% accuracy, 66.85% mIoU, and 74.03% DSC. The above two experiments demonstrate that unsupervised pre-training is an effective solution for tooth segmentation when the annotated data is severely limited. We can also notice that, with 40% labeled data, our method surprisingly achieves 88.27% mIoU, even surpassing all the supervised baselines trained with 100% labeled data in Tab. II.

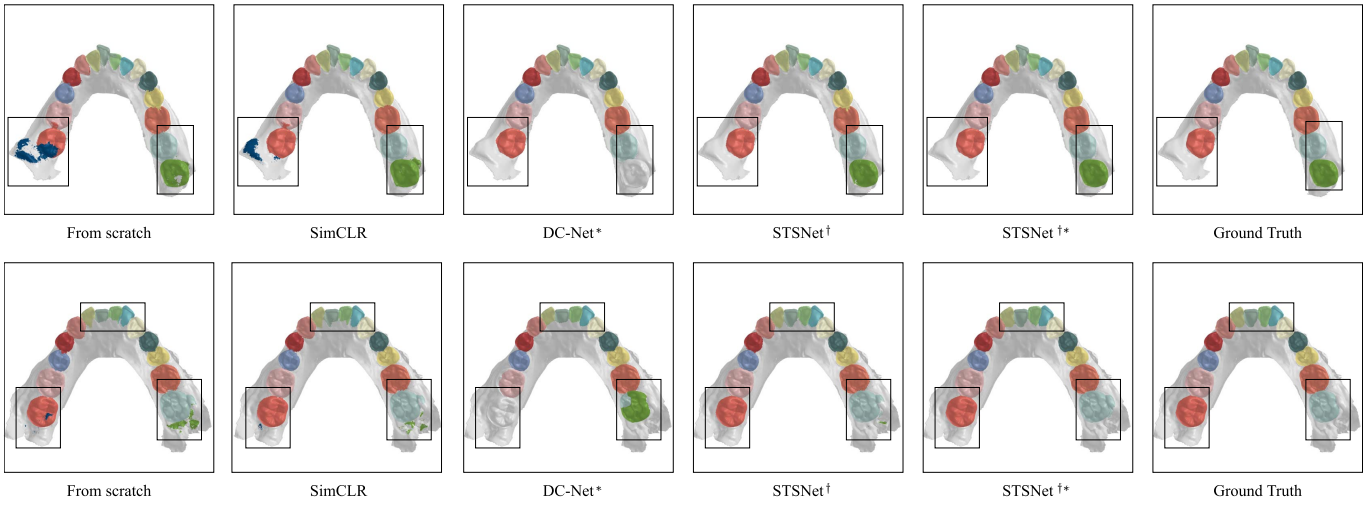

Fig. 5. Visualization of segmentation results of STSNet and baselines for two cases. See boxes for detailed comparison.

TABLE IV
STSNET PRETRAINED WITH DIFFERENT AUGMENTATIONS

Transformations	All		
	Acc	mIoU	DSC
translation	96.49(+0.67)	88.94(+2.39)	91.92(+1.92)
scale	96.40(+0.58)	88.49(+1.94)	91.58(+1.58)
rotation	96.76(+0.94)	89.48(+2.93)	92.20(+2.20)
translation + scale	96.44(+0.62)	89.42(+2.87)	92.41(+2.41)
translation + rotation	96.61(+0.79)	89.77(+3.22)	92.72(+2.72)

C. Ablation Studies

We conduct ablation experiments with several settings, including different transformations, loss functions and backbones to further understand the STSNet. Note that all ablation studies are conducted under 6,000 unlabeled data.

1) Ablation Study of Augmentation Strategies: Augmentation strategies usually have a non-trivial influence over the performance of unsupervised pre-training methods. Hence, we conduct an ablation study to quantitatively evaluate the effect of different augmentation methods during pre-training. The results are shown in Tab. IV. We can notice that rotation brings more remarkable improvement compared to translation and scale operations for unsupervised pre-training, which achieves 96.76% Acc, 89.48% mIoU and 92.20% DSC. When translation and rotation operations are simultaneously adopted, satisfactory performance is achieved, exceeding the supervised counterpart with 3.22% mIoU.

2) Ablation Study of Contrastive Losses: We also conduct ablation study on the coefficients in the contrastive loss. The results for different weighting coefficients in $\lambda\mathcal{L}^p + (1-\lambda)\mathcal{L}^r + \beta\mathcal{L}^c$ are shown in Tab. V. We can notice that the performance is not very sensitive to the weighting between point- and region-level losses, i.e., all of them are above 89% mIoU and 92% DSC without post-processing, which are better than the best-performing supervised DC-Net model, i.e., 87.34% mIoU and 90.67% DSC without graph-cut processing. Adding the cross-level loss $\mathcal{L}^c$ but the selection of β in the range of $[0.2, 0.5]$ has little influence to the overall performance.

TABLE V
ABLATIONS OF THE COEFFICIENTS IN CONTRASTIVE LOSSES

Setting	coefficients			All		
	$\mathcal{L}^p$	$\mathcal{L}^r$	$\mathcal{L}^c$	Acc	mIoU	DSC
from scratch	-	-	-	95.82	86.55	90.00
model-a	1	-	-	96.36	88.75	91.76
model-b	-	1	-	96.11	88.68	91.86
model-c	0.2	0.8	-	96.53	89.03	92.06
model-c	0.4	0.6	-	96.55	89.31	92.24
model-c	0.5	0.5	-	96.58	89.59	92.57
model-c	0.6	0.4	-	96.55	89.33	92.35
model-c	0.8	0.2	-	96.46	89.21	92.10
model-d	0.5	0.5	0.2	96.61	89.77	92.72

TABLE VI
SEGMENTATION PERFORMANCE ON DIFFERENT BACKBONES

Backbone	Mandible			Maxillary			All		
	Acc	mIoU	DSC	Acc	mIoU	DSC	Acc	mIoU	DSC
PointNet	87.52	68.23	76.09	91.26	75.18	81.56	89.65	72.19	79.21
DGCNN	95.25	84.31	88.18	96.25	88.25	91.37	95.82	86.55	90.00
DC-Net	95.32	84.81	88.52	96.40	89.26	92.28	95.94	87.34	90.67
STSNet(PointNet)	91.51	71.65	78.19	93.24	76.81	82.34	92.50	74.59	80.56
STSNet(DGCNN)	96.35	88.58	91.80	96.81	90.67	93.42	96.61	89.77	92.72

3) Ablation Study of the Backbone: We investigate the effectiveness of STSNet for different backbones with experiments on the widely-used PointNet and DGCNN models. The results are shown in Tab. VI. We can notice that our method can achieve 2.40% mIoU improvement compared to the supervised PointNet model. Meanwhile, the results of STSNet(DGCNN) over DGCNN and DC-Net also demonstrate the effectiveness of our method for DGCNN-like backbones. We also plot the training curve of for both backbones in two scenarios: training-from-scratch or supervised finetuning after pretrained with STSNet. We can find that backbones with STSNet converge much faster during training, as shown in Fig. 6. These results empirically demonstrate the consistent effectiveness of our method across different backbones.

D. Visualization

We demonstrate the superiority of STSNet† with case visualization in Fig. 5. The baselines are easy to commit

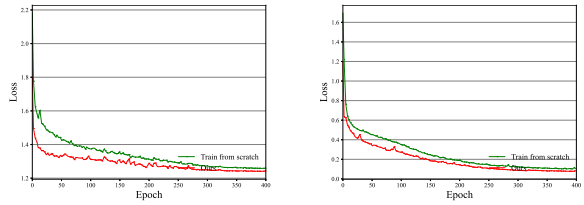

Fig. 6. The training loss curve with STSNet on different backbones during supervised fine-tuning. Left: DGCNN, Right: PointNet. Green and red curves indicate results for training-from-scratch and STSNet, respectively.

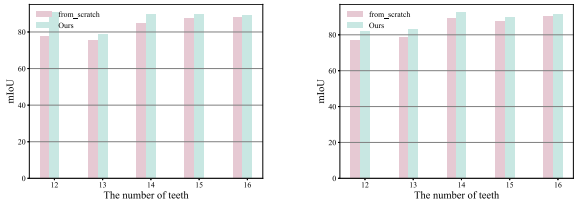

Fig. 7. Statistics of the performance w.r.t. number of tooth per IOS. (Left: mandible, Right: maxillary).

mistakes such as: (1) failing to recognize the molars, (2) failing to identify the tooth-tooth or tooth-gingiva boundaries; (3) misclassifying the tooth codes, (4) generating wrong and isolated tooth predictions for some small tooth parts. In stark contrast, the STSNet† seldomly commits such mistakes. Especially, the STSNet† can produce clear tooth predictions and rarely generates isolated tooth predictions, though many of these errors could be corrected by graph-cut smoothing. This is consistent with the superior performance of STSNet†* compared to DC-Net*.

V. Discussion

A. Statistical Analysis

We analyze the statistical performance of our method in each individual tooth to verify its effectiveness across diverse IOS scans, as each tooth might possess different anatomical features and boundaries. We also associate the corresponding missing ratios in our test set, the ratio of patients without the corresponding tooth, for detailed analysis. The results are reported in Tab. VII. Our method achieves consistent improvements over the baseline on all teeth, and achieves impressive improvements for many teeth, such as the first premolar and the third-molars (the 4-th and 8-th teeth). An interesting finding is that the improvements become more prominent for teeth with higher missing rate, e.g., an improvement of more than 10% mIoU for third-molars. This is maybe because our model learns better representations for the third-molar leveraging the large-scale unlabeled dataset that contains more corresponding samples.

Moreover, we also investigate the performance of our model in the IOS level with respective to different numbers of tooth in each IOS. The results are illustrated in Fig. 7. We can notice that our STSNet can achieve consistent performance improvement than the training-from-scratch supervised baseline in both maxillary and mandible across different numbers of teeth.

We further provide more random visualizations of the segmentation results of our method and the supervised counterpart, as shown in Fig. 8. We can see that our STSNet can better recognize the molars and the crowding adjacent teeth, is stable across different dental arches, and performs better for tooth-tooth/tooth-gingiva boundaries (case 1, 4) and for patients with dentural diastema (case 3) or erupted teeth (case 2, 3, 4), etc, demonstrating its effectiveness across various IOSs.

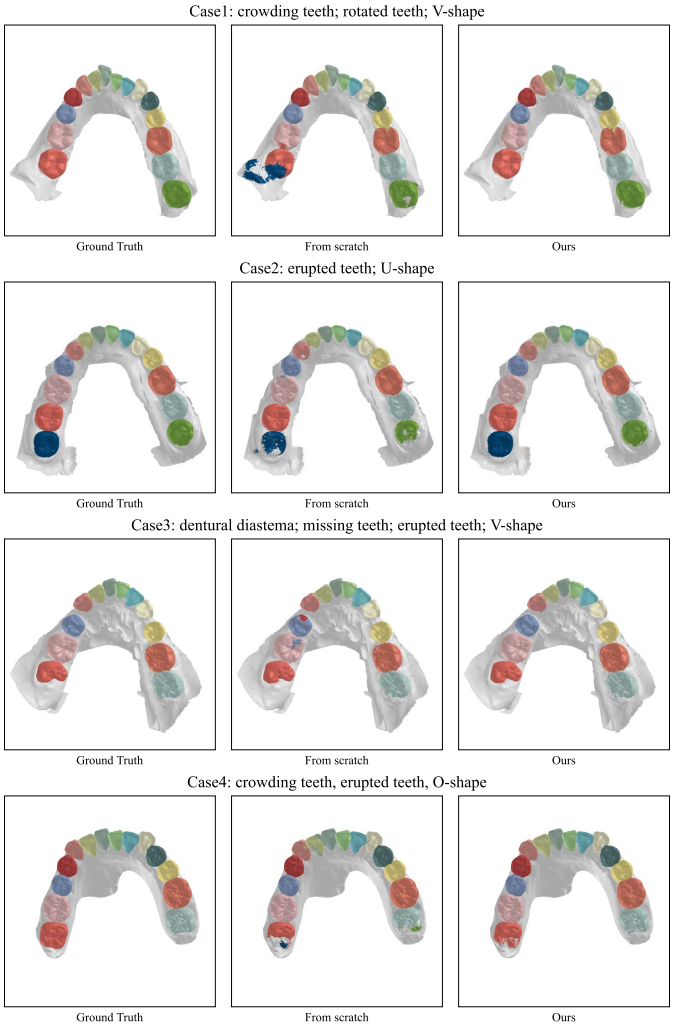

Fig. 8. Visualization of segmentation results on diverse IOSs. Top to bottom: case 1, 2, 3 and 4.

B. Effectiveness of Dynamic Graphs in the Backbone

We conduct experiments to find how the dynamic graphs along the DGCNN backbone contribute to the final performance. Particularly, we do experiments with two different architectures: 1) DGCNN with dynamic graphs; 2) DGCNN without dynamic graphs, i.e., computing the nearest neighbors based on the 3D coordinates. Results are reported in Tab. VIII. In general, DGCNN with dynamic graphs outperforms the counterpart without dynamic graphs in both supervised and unsupervised settings, i.e., an improvement of 0.93% and 1.92% on mIoU.

TABLE VII
STATISTICAL PERFORMANCE ON INDIVIDUAL TOOTH OF THE TRAINING-FROM-SCRATCH BASELINE/STSNET† ON THE TEST SET (200 PATIENTS). NUMBERS IN THE BRACES DENOTE THE PERFORMANCE GAIN OF STSNET†

Tooth	11	12	13	14	15	16	17	18
Missing rate (%)	0.00	0.88	0.88	7.02	0.00	0.00	0.00	72.81
mIoU	92.13/94.03(+1.90)	89.51/91.95(+2.44)	91.98/93.07(+1.09)	87.23/89.91(+2.68)	92.20/93.72(+1.52)	92.60/94.10(+1.50)	91.64/92.45(+0.81)	71.73/82.35(+10.62)
Tooth	21	22	23	24	25	26	27	28
Missing rate (%)	0.00	0.00	0.00	9.65	1.75	1.75	0.00	69.30
mIoU	92.27/93.34(+1.07)	90.63/92.39(+1.76)	91.11/93.50(+2.39)	83.25/94.81(+11.56)	89.39/89.63(+0.24)	88.50/90.46(+1.96)	88.66/90.24(+1.58)	72.19/84.84(+12.65)
Tooth	31	32	33	34	35	36	37	38
Missing rate (%)	5.81	4.65	0.00	12.79	1.16	0.00	0.00	67.44
mIoU	78.67/80.57(+1.90)	83.14/85.76(+2.62)	89.79/91.68(+1.89)	78.54/87.81(+9.27)	89.84/90.45(+0.61)	89.89/91.05(+1.16)	87.50/89.21(+1.71)	72.40/86.07(+13.67)
Tooth	41	42	43	44	45	46	47	48
Missing rate (%)	4.65	6.98	0.00	13.95	1.16	0.00	0.00	65.12
mIoU	78.65/82.85(+4.20)	80.73/84.18(+3.45)	89.11/91.22(+2.11)	80.22/91.45(+11.23)	90.33/91.87(+1.54)	90.49/92.29(+1.80)	87.03/89.24(+2.21)	73.05/82.94(+9.89)

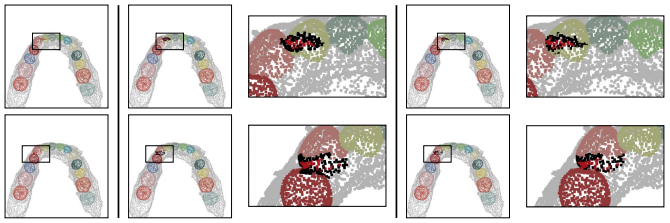

Fig. 9. Visualization of the learned neighbors in feature spaces produced by backbones with or without dynamic graphs. Red: Anchor points; Black: Learned neighbor points.

TABLE VIII
SEGMENTATION PERFORMANCE FOR BACKBONES WITH/WITHOUT DYNAMIC GRAPHS

Method	Train Strategy	Mandible			Maxillary			All		
		Acc	mIoU	DSC	Acc	mIoU	DSC	Acc	mIoU	DSC
w/o dynamic graphs	from scratch	94.42	83.30	87.61	95.43	87.37	90.87	95.00	85.62	89.47
	Our	95.37	84.96	88.54	96.65	90.03	92.79	96.10	87.85	90.97
w dynamic graphs	from scratch	95.25	84.31	88.18	96.25	88.25	91.37	95.82	86.55	90.00
	Our	96.35	88.58	91.80	96.81	90.67	93.42	96.61	89.77	92.72

A more straightforward visualization of the selected neighbors in the backbone is illustrated in Fig. 9. We randomly pick a set of anchor points and visualize their corresponding nearest neighbors in the feature space. We can notice that the neighbors selected by the network without dynamic graph are less accurate than the network with dynamic graph, i.e., it might wrongly select neighbors from adjacent molars. These results demonstrate the effectiveness of using dynamic graphs along the backbone.

C. Exploration of Semi-Supervised Methods

We explore the self-supervised learning methods on our dataset by experiments with the widely-used Self-training method [55], [56]. It includes three steps without the need of iterative training: **1) Supervised Learning:** Train a teacher model T on $\mathcal{D}^l$ with cross-entropy loss. **2) Pseudo Labeling:** Predict one-hot hard pseudo labels on $\mathcal{D}^u$ with T to obtain $\hat{\mathcal{D}}^u = \{(u_i, T(u_i))\}_{i=1}^N$. **3) Re-training:** Re-train a student model S on the union set $\mathcal{D}^l \cup \hat{\mathcal{D}}^u$. We also try to combine our method with general semi-supervised methods. Following the setting in SimCLR v2 [57], which includes three steps without iterative training: (1) unsupervised or self-supervised pre-training, (2) supervised fine-tuning, and (3) distillation using unlabeled data, we first apply our STSNet on 6,000 unlabeled 3D IOS data in pre-training and 600 labeled data in fine-tuning to obtain a tooth segmentation network. Then we use the fine-tuned network as a teacher to impute labels for training a student network (we also combine the labeled data in training student network as suggest in SimCLR v2). The student network is used for the final tooth segmentation.

TABLE IX
RESULTS WITH SEMI-SUPERVISED LEARNING

Method	Mandible			Maxillary			All		
	Acc	mIoU	DSC	Acc	mIoU	DSC	Acc	mIoU	DSC
Supervised	95.25	84.31	88.18	96.25	88.25	91.37	95.82	86.55	90.00
Semi-Supervised	94.89	85.82	89.80	96.23	89.97	93.05	95.65	88.19	91.65
Semi-Supervised with STSNet	96.35	88.36	91.56	96.89	91.18	93.81	96.66	89.97	92.84
STSNet	96.35	88.58	91.80	96.81	90.67	93.42	96.61	89.77	92.72

The preliminary results are reported in Tab. IX. We can note that, under the same data budget (6,000 unlabeled and 1,000 labeled data), a simple self-training scheme achieves 1.64% mIoU improvement to the supervised counterpart, though it is still a bit worse than our specifically-designed self-supervised strategy. Using STSNet to generate pseudo label and re-training achieves 3.42% mIoU improvement compared to the supervised counterpart, further exceeding our STSNet by 0.2% mIoU, which is consistent with the results in SimCLR v2. We will further explore this based on such promising results.

D. Effectiveness of STSNet on Public Dataset

We conduct experiments on the ShapeNetPart [46], a public 3D point cloud dataset. We further compare our results to Point-BERT [44] (following the same experimental settings, we pretrain our model on ShapeNet [58], and then finetune it on ShapeNetPart.) Please note that Point-BERT [44] is based on generative self-supervised learning but not contrastive learning, and it is later than our work. The results are reported in Tab. X. We also include the PointContrast [19] for more comprehensive comparison, as there are very few works for 3D self-supervised learning on point cloud, though it is based on voxel method. We can notice that PointBERT, achieves a 0.5% performance gain, and PointContrast reports a 0.4% performance gain. Our STSNet also achieves competing performance with 0.5% performance gain, demonstrating its effectiveness on other public dataset.

Furthermore, we investigate the effect of pre-training with different amounts of unlabeled data, with results in Tab. XI. We also introduce ModelNet40 [58] (contains 12,311 clean 3D CAD models) to our pretraining dataset, which is used

TABLE X
SEGMENTATION PERFORMANCE (MIOU) ON SHAPENETPART OF STSNET AND BASELINES. 'FROM SCRATCH' DENOTES THE PERFORMANCE WITHOUT UNSUPERVISED PRE-TRAINING STRATEGY; 'W PRE-TRAINING' REPRESENTS THE PERFORMANCE WITH UNSUPERVISED PRE-TRAINING; Δ DENOTES THE PERFORMANCE GAIN; 'DATA FORMAT' DENOTES THE DATA (VOXEL/POINT) DURING TRAINING

Method	from scratch	w pre-training	Δ	data format
PointContrast [19]	84.7	85.1	0.4	voxel
Point-BERT [44]	85.1	85.6	0.5	point
STSNet	85.5	86.0	0.5	point

TABLE XI
SEGMENTATION PERFORMANCE (MIOU) OF STSNET PRE-TRAINED WITH DIFFERENT AMOUNTS OF DATA (PUBLIC DATASET)

Data	from scratch	w pre-training	Δ
10,000 ShapeNet data	85.50	85.79	0.29
20,000 ShapeNet data	85.50	85.95	0.45
All ShapeNet data	85.50	85.99	0.49
All ShapeNet and ModelNet40 data	85.50	86.03	0.53

to verify the effectiveness of extra data. In public dataset ShapeNet, when using 10,000 data during pre-training, our model achieves 0.29% performance gain. As the amount of data increases gradually, the performance is further improved, finally reaching 0.49%performance gain. With the addition of other dataset ModelNet40, our model can only get a slight boost, about 0.04%. The possible reason for this little improvement may be that the public 3D object data are relatively simple (man-made), and the ModelNet dataset is quite different from ShapeNet, i.e., regarding as out-of-distribution (OOD) data. Though some recent research suggest that OOD data could also be leveraged for better self-supervised long-tailed learning in 2D images [59], a better design is yet under development to effectively integrate OOD data in the pre-training for 3D data.

E. Inference Strategy and Post-Processing

We further demonstrate the segmentation results when inferred with different amounts of points before kNN and graph-cut smoothing, as shown in Fig. 10. With the increasing number of inference points, both training from scratch and STSNet will will have a performance improvement, e.g., STSNet can further achieve 90.60% mIoU with 80,000 point with 0.72% improvement over result in Tab. I. Meanwhile, graph-cut smoothing can boost the performance of both DC-Net and STSNet, mainly by refining the boundaries and correcting isolated predictions.

F. Limitation

There are nevertheless some limitations of our work. First, as the first attempt for self-supervised 3D tooth segmentation, our work can be further improved in terms of the self-supervised framework. More augmentation strategies, architectures and loss functions could be explored for better performance. The computational complexity of self-supervised learning method could also be investigated to develop more

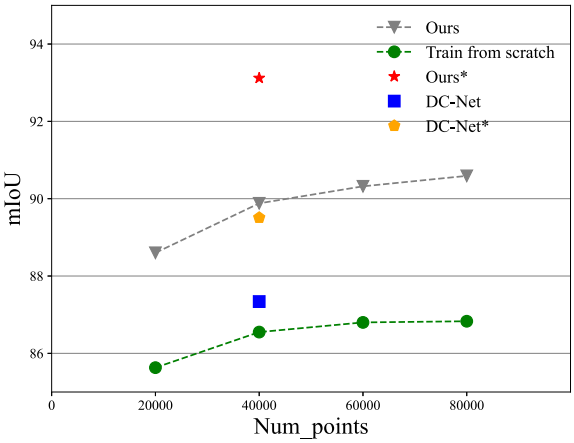

Fig. 10. The segmentation result of several methods with different points during post-processing.

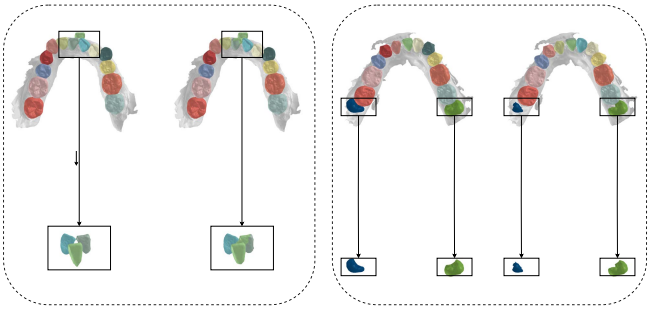

Fig. 11. Visualization of errors in incisors.

efficient learning paradigms. Second, the geometrical and topological features in the original mesh scans are not fully exploited either with supervised or unsupervised strategies for tooth segmentation. Finally, the robustness and generalization ability across complicated cases could be improved. Based on our statistical analysis, we can find that our method can still be improved a lot, especially for some hard teeth, such as incisors or third-molar. In fact, there would always be hard IOS cases that cannot be resolved by end-to-end deep learning models, hence, the generalization ability, or moreover, the interpretability of such 3D tooth segmentation systems need to be considered in the future, especially with the goal of deploying clinically applicable solutions for diagnosis and treatment planning.

VI. CONCLUSION

In this paper, we propose the first unsupervised pre-training framework with three hierarchical level contrastive learning loss functions for 3D tooth segmentation in introoral mesh scans. The extensive experiments convincingly corroborate the effectiveness of the proposed unsupervised pre-training strategy for helping alleviate the necessity of large-scale labeled training data for accurate 3D tooth segmentation. In future work, we will further investigate more advanced self-supervised learning strategies for better representation learning, e.g., adding additional variance or covariance regularizations [60] in point or region embeddings. We will also investigate advanced semi-supervised learning approaches,

which might help achieve improved performance over complicated IOS scans with heterogeneous anatomical features for clinically applicable diagnosis.

REFERENCES

[1] K. Wu, L. Chen, J. Li, and Y. Zhou, "Tooth segmentation on dental meshes using morphologic skeleton," *Comput. Graph.*, vol. 38, no. 1, pp. 199–211, 2014.

[2] F. G. Zanjani et al., "Mask-MCNet: Instance segmentation in 3D point cloud of intra-oral scans," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2019, pp. 128–136.

[3] F. G. Zanjani et al., "Deep learning approach to semantic segmentation in 3D point cloud intra-oral scans of teeth," in *Proc. Int. Conf. Med. Imag. Deep Learn.*, 2019, pp. 557–571.

[4] D. Sun et al., "Tooth segmentation and labeling from digital dental casts," in *Proc. IEEE 17th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2020, pp. 669–673.

[5] J. Zhang, C. Li, Q. Song, L. Gao, and Y.-K. Lai, "Automatic 3D tooth segmentation using convolutional neural networks in harmonic parameter space," *Graph. Models*, vol. 109, May 2020, Art. no. 101071.

[6] T.-H. Wu et al., "Two-stage mesh deep learning for automated tooth segmentation and landmark localization on 3D intraoral scans," 2021, *arXiv:2109.11941*.

[7] T. Yuan, W. Liao, N. Dai, X. Cheng, and Q. Yu, "Single-tooth modeling for 3D dental model," *Int. J. Biomed. Imag.*, vol. 2010, pp. 1–14, Dec. 2010.

[8] W. Herrmann, "On the completion of federation dentaire internationale specifications," *Zahnarztliche Mitteilungen*, vol. 57, no. 23, pp. 1147–1149, 1967.

[9] J. Hao et al., "Toward clinically applicable 3-dimensional tooth segmentation via deep learning," *J. dental Res.*, vol. 101, pp. 304–311, Mar. 2021.

[10] X. Xu, C. Liu, and Y. Zheng, "3D tooth segmentation and labeling using deep convolutional neural networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 7, pp. 2336–2348, Jul. 2018.

[11] S. Tian, N. Dai, B. Zhang, F. Yuan, Q. Yu, and X. Cheng, "Automatic classification and segmentation of teeth on 3D dental model using hierarchical deep learning networks," *IEEE Access*, vol. 7, pp. 84817–84828, 2019.

[12] C. Lian et al., "Deep multi-scale mesh feature learning for automated labeling of raw dental surfaces from 3D intraoral scanners," *IEEE Trans. Med. Imag.*, vol. 39, no. 7, pp. 2440–2450, Jul. 2020.

[13] Z. Cui et al., "TSegNet: An efficient and accurate tooth segmentation network on 3D dental model," *Med. Image Anal.*, vol. 69, Apr. 2021, Art. no. 101949.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[15] J.-B. Grill et al., "Bootstrap your own latent: A new approach to self-supervised learning," 2020, *arXiv:2006.07733*.

[16] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[18] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Jun. 2021, pp. 9782–9792.

[19] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "PointContrast: Unsupervised pre-training for 3D point cloud understanding," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 574–591.

[20] L. Liu, A. I. Aviles-Rivero, and C.-B. Schönlieb, "Contrastive registration for unsupervised medical image segmentation," 2020, *arXiv:2011.08894*.

[21] A. Taleb, M. Kirchler, R. Monti, and C. Lippert, "ContIG: Self-supervised multimodal contrastive learning for medical imaging with genetics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 20908–20921.

[22] K. Chaitanya, E. Erdil, N. Karani, and E. Konukoglu, "Contrastive learning of global and local features for medical image segmentation with limited annotations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12546–12558.

[23] D. Zeng et al., "Positional contrastive learning for volumetric medical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2021, pp. 221–230.

[24] B. Dufumier et al., "Contrastive learning with continuous proxy meta-data for 3D MRI classification," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2021, pp. 58–68.

[25] J. Li et al., "Multi-task contrastive learning for automatic CT and X-ray diagnosis of COVID-19," *Pattern Recognit.*, vol. 114, Jun. 2021, Art. no. 107848.

[26] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8895–8904.

[27] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Nov. 2019.

[28] X. He et al., "Unsupervised pre-training improves tooth segmentation in 3-dimensional intraoral mesh scans," in *Proc. Int. Conf. Med. Imag. With Deep Learn.*, 2022.

[29] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, vol. 1, 2014, pp. 766–774.

[30] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1920–1929.

[31] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1422–1430.

[32] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning?" in *Proc. 13th Int. Conf. Artif. Intell. Statist.* JMLR Workshop and Conference Proceedings, 2010, pp. 201–208.

[33] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6706–6716.

[34] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[35] Y. Zhang, R. He, Z. Liu, K. H. Lim, and L. Bing, "An unsupervised sentence embedding method by mutual information maximization," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 1601–1610.

[36] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 6894–6910.

[37] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[38] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[39] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," 2017, *arXiv:1706.02413*.

[40] H. Ran, W. Zhuo, J. Liu, and L. Lu, "Learning inner-group relations on point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15477–15487.

[41] T. Xiang, C. Zhang, Y. Song, J. Yu, and W. Cai, "Walk in the cloud: Learning curves for point clouds shape analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 915–924.

[42] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16259–16268.

[43] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu, "Spatio-temporal self-supervised representation learning for 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6535–6545.

[44] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19313–19322.

[45] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[46] L. Yi et al., "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, 2016.
[47] T. Kondo, S. H. Ong, and K. W. C. Foong, "Tooth segmentation of dental study models using range images," *IEEE Trans. Med. Imag.*, vol. 23, no. 3, pp. 350–362, Mar. 2004.
[48] Z. Li, X. Ning, and Z. Wang, "A fast segmentation method for STL teeth model," in *Proc. IEEE/ICME Int. Conf. Complex Med. Eng.*, May 2007, pp. 163–166.
[49] Y. Kumar, R. Janardan, B. Larson, and J. Moon, "Improved segmentation of teeth in dental models," *Comput.-Aided Des. Appl.*, vol. 8, no. 2, pp. 211–224, 2011.
[50] R. Fan, X. Jin, and C. C. L. Wang, "Multiregion segmentation based on compact shape prior," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 1047–1058, May 2015.
[51] Z. Li and H. Wang, "Interactive tooth separation from dental model using segmentation field," *PLoS ONE*, vol. 11, no. 8, Aug. 2016, Art. no. e0161159.
[52] H. Deng, T. Birdal, and S. Ilic, "PPFNet: Global context aware local features for robust 3D point matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 195–205.
[53] H. Deng, T. Birdal, and S. Ilic, "PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 602–618.
[54] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, "D3Feat: Joint learning of dense detection and description of 3D local features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6359–6367.
[55] D.-H. Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Represent. Learn. (ICML)*, 2013, vol. 3, no. 2, p. 896.
[56] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "ST++: Make self-training work better for semi-supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4268–4277.
[57] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 22243–22255.
[58] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.
[59] H. Wei, L. Tao, R. Xie, L. Feng, and B. An, "Open-sampling: Exploring out-of-distribution data for re-balancing long-tailed datasets," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 23615–23630.
[60] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-invariance-covariance regularization for self-supervised learning," 2021, *arXiv:2105.04906*.