Computers & Graphics 132 (2025) 104313

Contents lists available at ScienceDirect

Computers & Graphics

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

Special Section on SMI 2025

Learning center- and boundary-aware instance representation for 3D tooth segmentation

Hairong Jin [a,b], Jianwen Lou [c], Zhiguo Lu [c], Teng Wu [d], Kun Zhou [a], Youyi Zheng [a,*]

[a] State Key Lab of CAD&CG, Zhejiang University, Hangzhou, 310058, China
[b] School of Computer and Computing Science, Hangzhou City University, Hangzhou, 310015, China
[c] School of Software Technology, Zhejiang University, Ningbo, 315100, China
[d] Lightspeed Studios, Tencent, Shenzhen, 518057, China

ARTICLE INFO

Keywords:
3D tooth segmentation
Geodesic distance
Center-aware sampling
Interactive refinement

ABSTRACT

Automatic and precise 3D tooth instance segmentation is crucial in digital stomatology. Despite breakthrough progress by learning-based methods, atypical cases still pose significant challenges for tooth semantics and boundary discrimination. We argue that one of the primary reasons for this is the limited expressiveness of the feature representations of 3D tooth instances learned by deep neural networks. To address this issue, we introduce a novel segmentation framework that features a neural network capable of learning center- and boundary-aware tooth instance representations. Our network begins with an encoder that extracts geometric features from the 3D tooth mesh, supervised by semantic-agnostic boundary and distance maps. Leveraging an effective center-aware sampling strategy based on the distance map, we select the representative candidate points located around the tooth centers and utilize local graph convolution and candidate-wise self-attention to learn instance representation. These candidate features are then combined with point-wise features and processed through prediction heads, which generate corresponding candidate masks, class probabilities, and IoU scores in parallel. Finally, instance masks are selected and merged via non-maximum suppression (NMS) to produce the final segmentation results. Additionally, to maximize the optimization of segmentation results, we design an interactive refinement scheme that allows users to select desired regions with handy clicks and correct erroneous labels directly on the high-resolution original tooth mesh. We extensively validate our method on the Teeth3DS benchmark dataset, demonstrating superior performance in tooth category and boundary identification compared to state-of-the-art methods.
(The code well be released at https://github.com/bibi547/CBANet.git).

1. Introduction

With advancements in scanning and reconstruction technology, obtaining high-resolution 3D tooth models from various sources like intraoral scanners (IOS) and cone-beam computed tomography (CBCT) systems [1] has become increasingly feasible. This progress supports the broad implementation of computer-aided design (CAD) systems in orthodontic treatments [2]. Within digital orthodontic diagnostics and therapy, 3D tooth segmentation is a crucial step that aims to automatically isolate each tooth from the jaw mesh and assign it the correct index based on semantic labels. Precise segmentation significantly aids dentists in treatment simulation, orthodontic planning, and outcome evaluation, among other clinical applications. Despite numerous proposed methods [3–5] for analyzing and processing 3D tooth models, achieving both accurate and efficient segmentation presents ongoing challenges. This is primarily due to the diversity in patient tooth cases, such as crowding and missing teeth, which make it difficult for segmentation algorithms to locate teeth and distinguish boundaries. Additionally, noise in intraoral scan data and dental attachments further complicate the segmentation process.

Recent breakthroughs in deep learning for 3D vision and geometry processing have opened up new avenues for advancing the performance of 3D tooth segmentation algorithms. Most existing deep learning-based methods [6–8] strive to predict semantic labels for 3D point clouds or meshes directly, leveraging the network's capacity for extracting geometric features. However, learning-based methods lack stability, inevitably leading to isolated erroneous predictions and inaccurate boundaries. To make up for these deficiencies as much as possible, numerous tooth segmentation approaches focus on the distinction of the features around tooth boundaries. Examples include

* Corresponding author.
E-mail addresses: hairong_jin@zju.edu.cn (H. Jin), youyizheng@zju.edu.cn (Y. Zheng).

https://doi.org/10.1016/j.cag.2025.104313
Received 20 June 2025; Accepted 14 July 2025
Available online 6 August 2025
0097-8493/© 2025 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

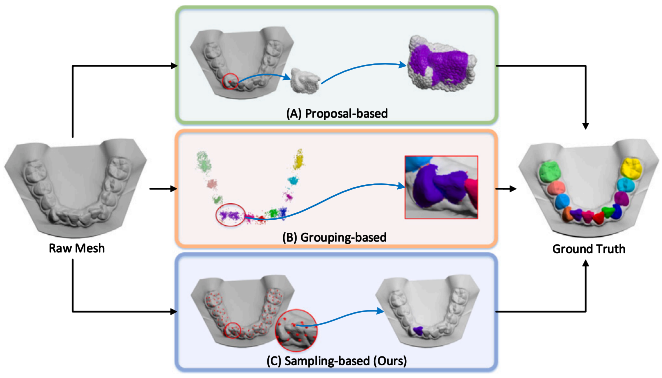


Fig. 1. Pipelines of various types of 3D tooth segmentation methods. A: Proposal-based methods heavily depend on the quality of 3D patches. B: Grouping-based methods may mistakenly cluster closely spaced teeth. C: Our method generates tooth masks by sampling candidate points.

supervising tooth boundaries through an auxiliary branch [9] or designing distinct streams to learn the high-dimensional representations of coordinates and normals [10]. Furthermore, the application of contrastive learning [11,12] aims to amplify the differentiation between features across tooth boundaries. Despite the progress that has been made, existing methods still struggle to accurately identify tooth boundaries, highlighting deficiencies in the intrinsic feature learning of tooth instances.

In contrast, the existing state-of-the-art methods can be categorized into proposal-based [13,14], grouping-based [15–17], and sampling-based [18,19]. These methods explicitly represent instances in various manners, as illustrated in Fig. 1. Proposal-based [20–23] methods adopt a top-down 3D object cropping strategy, initially predicting a bounding box or a center for each tooth, then segmenting individual teeth from the cropped patches. While these methods simplify the segmentation task, the performance of instance masks is significantly influenced by the qualities of the cropped patches. Grouping-based [17] methods heavily depend on prior knowledge for hyperparameter tuning, resulting in limited fault tolerance and generalization capabilities. Inappropriate hyperparameter settings of the clustering algorithm can mistakenly cluster closely spaced teeth together or separate large-scale teeth, leading to fused or halved teeth. Alternatively, a few point cloud instance segmentation methods have introduced the concept of generating candidate masks via sampled points [18], while relying solely on the farthest point sampling (FPS) algorithm falls short in ensuring recall. This limitation is primarily attributed to the diversity in tooth sizes and the presence of cases with misaligned and crowded teeth. Even when employing instance-aware FPS [19], achieving the optimal sampling for representative candidates remains a challenge. This issue arises from the spatial structure of 3D tooth models, as the teeth align closely along the dental arch curve [23,24]. Consequently, these point sampling methods tend to preferentially sample points near the tooth boundaries. Moreover, high-quality instance feature learning necessitates the application of boundary constraints to the feature aggregation of candidate points.

To address these challenges, we introduce a novel 3D tooth segmentation framework, combining an instance segmentation network with an interactive refinement scheme. The center- and boundary-aware instance segmentation network, named CBANet, primarily features a center-aware farthest point sampling strategy to generate tooth masks. Initially, the encoder of CBANet constructed from stacked static EdgeConv modules [17] is applied to extract geometric features. To enhance the representation of candidates for their corresponding tooth instances and reduce ambiguous candidates, we aim to sample points as far from the boundaries as possible while avoiding excessive proximity to each other. Motivated by these considerations, we first introduce two

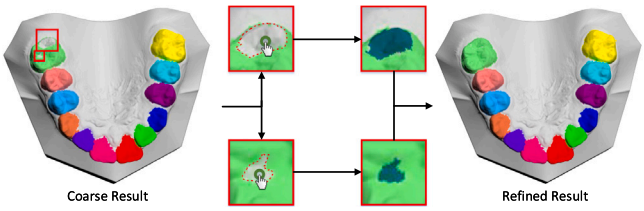


Fig. 2. Illustration of our interactive refinement scheme. Users can correct mispredicted regions with a few clicks. The red dashed line indicates the regions the user intends to correct, while the dark green areas represent the regions refined by our scheme.

dedicated decoder branches to supervise the category-agnostic boundary and distance maps, thereby enhancing the network's boundary-awareness capability. Leveraging the geodesic distance map, our well-designed center-aware farthest point sampling strategy ensures high recall rates while sampling high-quality candidate points. Then, we incorporate a feature aggregation module to encode candidate instance features, which combines local graph convolution and a self-attention block to capture both inter-point and inter-candidate relationships. The ultimate prediction heads process the candidate and mask features to generate candidate masks, category probabilities, and IoU scores. Since each sampled point naturally corresponds to a specific tooth, CBANet inherently supports a streamlined end-to-end training pipeline. During inference, we select the optimal mask for each tooth instance from the candidate masks based on predicted category probabilities and IoU scores, thereby merging them to generate the final segmentation results. Additionally, to address some of the inevitable prediction errors, we provide users with an interactive refinement scheme, as shown in Fig. 2. Starting from the user's click points, the optimization strategy iteratively expands toward the desired boundaries, enabling precise selection of the target region. This scheme allows users to directly select large erroneous patches on the original high-resolution model and correct them with the specified labels.

We conduct thorough experiments and analyses on the large-scale 3D tooth benchmark dataset, Teeth3DS [25]. The experimental results demonstrate that CBANet outperforms existing methods in both qualitative and quantitative comparisons. Furthermore, our interactive refinement tool efficiently identifies user-specified error regions, enabling quick correction of mispredicted regions.

In summary, our contributions are listed as follows:

- We present CBANet, an innovative network for 3D tooth segmentation, utilizing the geodesic distance map for center-aware farthest point sampling to learn instance representation.
- We design an interactive refinement scheme that allows users to select and correct the desired error regions on the initial segmentation results of the original tooth meshes with simple clicks.
- We evaluate the effectiveness of CBANet and the interactive refinement tool on the Teeth3DS benchmark through various performance metrics.

2. Related work

2.1. 3D tooth segmentation

Tooth segmentation is a fundamental application of 3D shape segmentation in computer-aided design systems for dental clinical treatments. In the wake of deep learning's ascendancy, data-driven methods are increasingly adopted for 3D segmentation, resulting in a significant performance gap compared to traditional methods. For example, Xu et al. [26] convert handcrafted geometric features of facets into 2D images and then process these feature maps with convolutional neural networks. TSGCNet [10] keenly observes the geometric feature variations

across the surfaces of 3D tooth models, thereby designing a dual-stream graph convolutional network to separately process coordinates and normals, achieving higher precision at tooth boundaries. To streamline segmentation tasks and improve tooth segmentation accuracy, several proposal-based methods, inspired by Mask R-CNN [27], have been introduced [13,14,28]. These methods employ a top-down pipeline for segmenting instances within point clouds. Mask-MCNet [22], initially identifies each tooth bounding box before proceeding to its segmentation, although this two-stage method tends to be less efficient. TSegNet [21] determines each tooth's position by predicting its centroid, followed by segmentation within 3D crops. Introducing a weakly supervised strategy, DArch [23] predicts dental arches to pinpoint tooth centroids, facilitating the segmentation of tooth instances from weak annotations. Nevertheless, segmenting instances from cropped patches neglects the global context information of teeth within jaw models. Alternatively, Grouping-based methods [15,16,29,30] suggest a strategy of predicting offsets to vote centroids for clustering. Yet, these bottom-up approaches typically require clustering algorithms to aggregate homogeneous points. TeethGNN [17] clusters tooth points by predicting offsets and semantic labels. While it achieves high efficiency and accuracy, it necessitates intricate post-processing operations, such as clustering and voting, requiring substantial prior knowledge for hyperparameter tuning. PointInst3D [18] employs farthest point sampling to generate instance masks, considering sampled points as candidates. However, simple FPS struggles to ensure that sampled points cover all instances. Additionally, the issue is compounded by the diverse scales of instances, leading to a non-ideal distribution of sampled points with an excess typically found in background areas. ISBNet [19] proposes an instance-aware sampling strategy combined with the prediction of 3D bounding boxes to capture instance scale as a crucial geometric feature. Instance-aware farthest point sampling inevitably tends to prioritize points located on the boundaries, a drawback that becomes particularly pronounced in tooth models where teeth are closely arranged along the dental arch. Therefore, we aim to prioritize sampling points that are far from the tooth boundaries to obtain more representative candidates. Furthermore, to the best of our knowledge, we are the first to adopt a sampling-based method for generating candidates to predict tooth masks.

2.2. Interactive segmentation

Traditional interactive mesh segmentation methods [31,32] rely on handcrafted geometric features, which lack robustness and precision. Reliable interactive 3D tooth segmentation [33,34] requires highly detailed user input, such as constraint points and boundary annotation. Recently, with the advent of attention mechanisms, interactive segmentation in 2D images [35,36] has been widely explored. Users can assist the network in generating attention maps through sparse clicks, thereby selecting the influence regions for precise object segmentation. Kontogianni [37] first proposed an interactive object segmentation method based purely on 3D point data without relying on extra 2D images. However, this method often requires multiple positive or negative clicks to produce a high-quality object mask. Overall, fully relying on interactive 3D segmentation still places a significant burden on users. Thus, we consider it more practical to provide an interactive refinement tool after achieving high-accuracy instance segmentation, enabling users to correct a few mispredicted regions. TSRNet [38] employs the gradient information from geodesic distance maps to move the tooth boundaries toward predicted boundaries, effectively correcting mispredictions. However, this one-time global optimization process lacks controllability and is challenging to apply to high-resolution original models. Inspired by this, our interactive refinement scheme takes user click points and iteratively expands the inference range toward the tooth boundaries, allowing for the selection of the desired modification area.

3. Methodology

3.1. Overview

In this section, we elaborate on our two-stage 3D tooth segmentation framework, which consists of the center- and boundary-aware instance segmentation network CBANet and an interactive refinement scheme. Specifically, as illustrated in Fig. 3, the architecture of CBANet begins with a simple encoder composed of stacked static EdgeConv [8,17] modules that extract point-wise geometric features. Two distinct decoder branches predict the boundary and distance maps, respectively. Relying on the geodesic distance map, our center-aware farthest point sampling accurately samples candidate points for tooth instances to encode high-quality candidate instance features. These candidate features are utilized to decode the candidate masks, category probabilities, and IoU scores. During inference, we employ the NMS to select the optimal binary mask for each tooth instance. Ultimately, to correct a few erroneous prediction regions and maximize the optimization of segmentation results, we design an interactive refinement scheme. This allows users to select the desired region with minimal clicks and adjust its labels accordingly.

3.2. CBANet

3.2.1. Geometric feature extraction

Due to GPU memory limitations, we initially apply a mesh downsampling algorithm (such as QEM [39]) to reduce high-resolution original tooth meshes (with over 100,000 facets) down to 10,000 facets. This simplification facilitates efficient network training and inference processes.

Given a simplified tooth mesh with N facets, we define the network input as an $N \times 15$ matrix $\mathbf{P} \in \mathbb{R}^{N\times 15}$. Specifically, each facet is represented by a 15-dimensional feature vector, which includes the centroid coordinate ($N \times 3$), facet normal ($N \times 3$), and three offset vectors from the triangle vertices to the centroid ($N \times 9$). Considering the inherent topological structure of tooth meshes, we utilize static EdgeConv modules as the fundamental units in our point-wise encoder to aggregate features from local neighborhoods of Euclidean space. We stack five static EdgeConv [8,17] modules to broaden the receptive field of the cells. Subsequently, skip connections integrate shallow and deep feature maps. These fused features are fed into a Multilayer Perceptron (MLP) followed by a max pooling layer to extract the global embedding. The local and global feature maps are concatenated and then dimensionally reduced through an MLP to produce our point-wise geometric features $\mathbf{F} \in \mathbb{R}^{N\times D}$. Finally, two output heads, each consisting of two MLPs and a Conv1d layer, predict the boundary $\mathbf{M}_b \in \mathbb{R}^{N\times 3}$ (including tooth, boundary, and gingiva) and distance $\mathbf{M}_d \in \mathbb{R}^{N\times 1}$ maps, respectively.

3.2.2. Boundary and distance maps

Different from TSRNet [38], which iteratively adjusts tooth boundaries based on boundary and distance maps, we incorporate these elements into instance learning for tooth segmentation.

The auxiliary supervision of the boundary map potentially imposes boundary constraints during the process of highlighting tooth regions, producing masks with sharpened boundaries. Compared to point clouds, which necessitate the consideration of neighboring point labels, defining boundaries in meshes is more intuitive. A facet is considered a boundary facet if its adjacent facets carry different labels. We derive the ground truth boundary maps from the labels: the boundary facets are set to 1, the tooth facets to 2, and the gingiva facets to 0.

Centroid-based methods are extensively applied in 3D object detection and instance segmentation tasks. While they accurately pinpoint the positions of objects, they fail to capture critical information about instance boundaries and surface geometry. For example, relying solely on the Euclidean distance and offsets from points to the centroids is

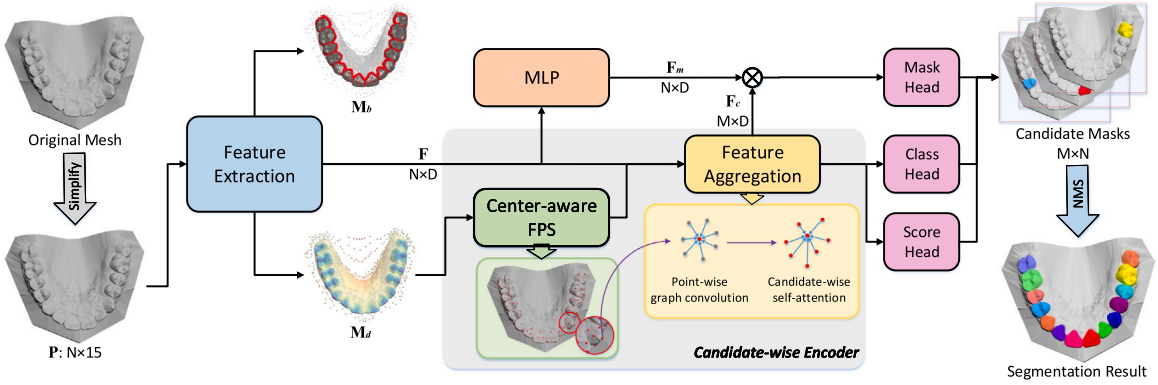


Fig. 3. The network architecture of CBANet.

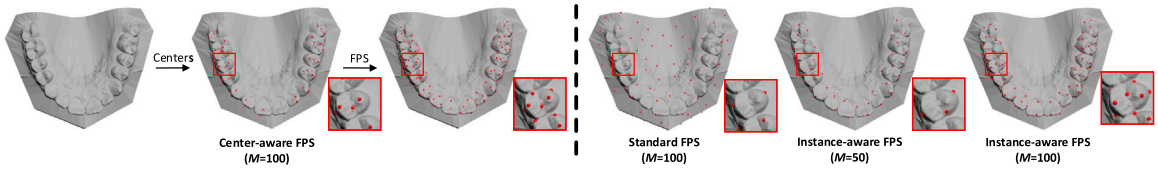


Fig. 4. Illustration of Center-aware FPS, Standard FPS, and Instance-aware FPS. Center-aware FPS strategically samples at the centers of teeth first, followed by FPS to capture the remaining candidates. In contrast, Standard FPS typically results in most sampled points within the gingiva, often neglecting small teeth. Instance-aware FPS prioritizes sampling points near tooth boundaries ($M = 50$). As the number of sampled points increases, progressively include points closer to the centers of the teeth ($M = 100$).

Table 1
The chamfer distance (CD) for various sampling methods at different sampled point counts on the Teeth3DS testing set.

	Upper					Lower				
	20	40	60	80	100	20	40	60	80	100
FPS	23.82	20.83	19.04	17.95	17.14	21.80	18.92	17.36	16.53	15.88
IA-FPS	12.28	9.98	8.95	8.56	8.21	11.73	9.31	8.26	7.89	7.58
CA-FPS	**4.49**	**3.90**	**3.90**	**3.87**	**3.87**	**4.30**	**3.57**	**3.56**	**3.54**	**3.52**

inadequate for effectively distinguishing between boundary and interior points within an instance. Compared to centroid-based offsets and heatmaps, geodesic distance maps focus more on boundary gradient information, enabling a more intuitive differentiation between the interior and boundary points. Firstly, we refer to facet centroids as points. To derive the ground truth geodesic distance map from semantic labels, we first calculate the geodesic distances from all points to boundary points, resulting in a boundary-based geodesic distance matrix. We then identify the minimum geodesic distance for each point. Finally, given the significant scale variation among tooth instances, we normalize the geodesic distance values within each instance and multiply the values of gingiva points by -1. These steps yield a geodesic distance map, which is normalized to $[-1, 1]$.

3.2.3. Candidate-wise encoder

The second phase is designed with two branches. The upper branch employs a straightforward MLP to process the boundary-aware features $\mathbf{F} \in \mathbb{R}^{N\times D}$ from the point-wise encoder, delivering mask features $\mathbf{F}_m \in \mathbb{R}^{N\times D}$. The lower branch is dedicated to learning instance representations by encoding candidate-wise features $\mathbf{F}_c$. It consists of two pivotal modules: center-aware farthest point sampling and a feature aggregation module. These modules are aimed at encoding candidate instance features $\mathbf{F}_c \in \mathbb{R}^{M\times D}$ from $\mathbf{F}$. Here, M denotes the preset number of candidates.

Center-aware FPS:

Farthest point sampling (FPS) iteratively selects points that are the furthest distance away from previously sampled points, thereby ensuring a uniform coverage of the entire 3D point set. However, relying exclusively on Euclidean distance for FPS when sampling candidate points may overlook smaller-scale objects, allowing backgrounds or larger-scale instances to dominate the majority of candidates. To achieve maximal coverage of all instances irrespective of their sizes, instance-aware FPS (IA-FPS) leverages semantic priors to sample the farthest points from a point set of differing semantics from previous candidates. This strategy tends to prioritize points near the instance boundaries, and only with a sufficiently sizeable preset number of candidates can it ensure the sampling of high-quality points closer to the instance centers. This issue becomes more pronounced in 3D tooth models, since tooth instances are aligned along the curvature of the dental arch.

Therefore, an ideal sampling strategy must not only cover all tooth instances but also selectively prioritize points distant from tooth boundaries to generate representative candidates. To this end, we develop a two-stage center-aware farthest point sampling method (CA-FPS) utilizing the geodesic distance map $\mathbf{M}_d$, as depicted in Fig. 4. Initially, we filter out points with predicted distance values below the lower threshold ($\epsilon_l = 0.2$), thus obtaining a filtered set of tooth points $\mathbf{S}_f$. In the first stage, we iteratively select center candidates. Specifically, we begin by choosing the point with the maximum distance value as a center candidate. Then, we use the k-nn to set the values of points within its local neighborhood to -1, updating the distance map. This selection process continues until the number of center candidates reaches $M/2$ or the maximum value in the updated geodesic distance map drops below a predefined upper threshold ($\epsilon_u = 0.5$), at which point we halt the iterations. In the second stage, we apply FPS starting from the selected center candidates on $\mathbf{S}_f$ to determine the remaining candidates.

The visualization of this process, as well as the comparisons with FPS and instance-aware FPS, are presented in Fig. 4. We observe that our sparse sampling set $\mathbf{S}_c = \{p_i\}_{i=1}^{M}$ not only encapsulates the

centers of each tooth but also ensures that each sampled point is significantly distanced from the tooth boundaries. However, standard FPS typically results in a majority of sampled points belonging to the gingival. In contrast, the sampled points of instance-aware FPS are predominantly located near the tooth boundaries. Consistently, we evaluate the chamfer distance between the sampled points and the tooth centers for various sampling methods at different sampled point counts, with results detailed in Table 1.

Feature Aggregation:

The feature aggregation module is composed of sequentially linked point-wise graph convolution and candidate-wise self-attention blocks. The point-wise graph convolution block aims to aggregate homogeneous point features within the local neighborhoods of the filtered set $\mathbf{S}_f$ for sampled candidates. It is noteworthy that $\mathbf{S}_f$ excludes points near tooth boundaries, effectively preventing cross-boundary local connections. Consequently, each candidate aggregates local features within its corresponding tooth instance, which facilitates the generation of high-quality candidate features. Specifically, for the feature $\mathbf{f}_i$ of each sampled candidate point $p_i \in \mathbf{S}_c$, the set of features within its k-nearest neighborhood in $\mathbf{S}_f$ is denoted as $\mathbf{F}^{(i)} = \{\mathbf{f}_{i,j}\}_{j=1}^{k} \subset \mathbf{F}$. The resulting output feature $\mathbf{f'}_i$ can be computed as follows:

$$\mathbf{f'}_i = \max_{j=1}^{k}(\mathrm{MLP}((\mathbf{f}_{i,j} - \mathbf{f}_i) \oplus \mathbf{f}_i)), \tag{1}$$

where $\oplus$ denotes the concatenation operation. By doing so, we obtain the candidate features $\mathbf{F'}_c = \{\mathbf{f'}_i\}_{i=1}^{M}$ after point-wise local aggregation.

Following this, the candidate-wise self-attention block is employed to facilitate information interchange among instances. Thus, the output candidate features $\mathbf{F}_c$ can be formulated as follows:

$$\mathbf{F}_c = \mathrm{Softmax}(\phi(\mathbf{F'}_c + \delta) \otimes \varphi(\mathbf{F'}_c)^{\top}) \otimes \psi(\mathbf{F'}_c + \delta) \tag{2}$$

$$\delta = \mathrm{MLP}(\mathrm{S}_c) \tag{3}$$

Here, δ is implemented as an MLP and is referred to as the position encoding, while ϕ, φ, ψ are realized using Conv1d. After the candidate-wise self-attention aggregation phase, we achieve a collection of M candidate features $\mathbf{F}_c \in \mathbb{R}^{M \times D}$.

3.2.4. Prediction heads

In the decoder phase, $\mathbf{F}_c$ is fed into a class head and a score head, which predict category probabilities and mask IoU scores, respectively. Additionally, given the mask $\mathbf{F}_m$ and candidate $\mathbf{F}_c$ features, the candidate mask features can be expressed as follows:

$$\mathbf{M'} = \mathbf{F}_c \otimes \mathbf{F}_m^{\top}, \tag{4}$$

where $\otimes$ denotes matrix production, and $\top$ represents the transpose. These features input the mask head to generate the candidate masks $\mathbf{M} \in \mathbb{R}^{M \times N}$.

All prediction heads consist of an MLP followed by a Conv1d layer. Furthermore, the mask head incorporates a sigmoid activation function.

3.2.5. Loss function

Since our candidate masks originate from sampled points, they inherently establish a many-to-one matching relationship with tooth instances in the mesh. We supervise these candidate masks using the masks of the instances to which the sampled points belong in the ground truth. This method obviates the need for the optimal bipartite matching process that relied on the Hungarian algorithm in previous approaches.

Our loss function is composed of point-wise loss and candidate-wise loss. Specifically, for the point-wise loss, we use cross-entropy loss to supervise the boundary map and L1 loss for the geodesic distance map. Here, λ_b and λ_d denote the weights for the boundary map loss and the distance map loss, respectively:

$$\mathcal{L}_e = -\lambda_b \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{3} g_{ij}^b \log y_{ij}^b + \lambda_d \frac{1}{N} \sum_{i=1}^{N} \left| y_i^d - g_i^d \right|, \tag{5}$$

where N represents the number of points. y_{ij}^b denotes the confidence of the jth category for the ith point in the predicted boundary map, while g_{ij}^b is the corresponding ground truth in one-hot format. y_i^d is the predicted value for the ith point in the geodesic distance map, and g_i^d is its ground truth.

The candidate-wise loss is comprised of mask loss $\mathcal{L}_m$, class loss $\mathcal{L}_c$, and score loss $\mathcal{L}_s$. The mask loss is a combination of BCE loss and Dice loss. We employ cross-entropy loss for class loss. Score loss is calculated using MSE loss. Notably, score loss is computed solely on positive samples. We treat a candidate as a positive sample if the IoU between the predicted mask and the ground truth exceeds 0.5.

$$\mathcal{L}_m = \frac{1}{M} \sum_{i=1}^{M} (\mathrm{BCE}(m_i, \hat{m}_i) + \mathrm{Dice}(m_i, \hat{m}_i)) \tag{6}$$

$$\mathcal{L}_c = \frac{1}{M} \sum_{i=1}^{M} \mathrm{CE}(c_i, \hat{c}_i) \tag{7}$$

$$\mathcal{L}_s = \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}_{\{\hat{s}_i > 0.5\}} \mathrm{MSE}(s_i, \hat{s}_i) \tag{8}$$

Here, M is the total number of sampled candidates. m_i, c_i, and s_i represent the predicted masks, probabilities, and scores for the ith candidate, respectively. $\hat{m}_i$, $\hat{c}_i$, and $\hat{s}_i$ correspond to the ground truth masks, one-hot encoded categories, and IoU scores.

Therefore, the overall loss function can be expressed as follows:

$$\mathcal{L} = \lambda_e \mathcal{L}_e + \lambda_m \mathcal{L}_m + \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s, \tag{9}$$

where the weights λ_e, λ_m, λ_c, λ_s serve as trade-off factors to balance the contributions of each respective loss.

3.2.6. Inference

Due to the close arrangement of teeth, intersections among predicted tooth masks are common. Challenging cases such as missing and crowded teeth frequently lead to semantic prediction errors. To effectively mitigate these failures, we implement a novel two-stage NMS strategy that considers both the confidence values and IoU scores of tooth masks. We initially select masks for each predicted tooth label based on the highest scores and confidence values, then combine these into a candidate set. This set is further filtered using three criteria to select the final masks: 1. A candidate mask's IoU with all other candidate masks is less than 0.5. 2. A mask's IoU with another mask of the same category is greater than 0.5 and possesses a higher IoU score. 3. A mask's IoU with a mask from a different category is greater than 0.5 and has a higher confidence value. Finally, we merge all selected masks to obtain the final segmentation results.

Unlike the post-processing operations in object detection tasks, our method involves fewer predicted candidate masks, and the selection and filtering of the candidate set are straightforward. As a result, our post-processing is rapid and does not burden the inference speed.

3.3. Interactive refinement

TSRNet [38] proposes an iterative refinement method that shifts boundaries toward the target positions using gradient information from the distance map. However, this method is limited to simplified meshes and lacks controllability. The refined results partly depend on the initial coarse segmentation quality. For example, as shown in Fig. 6, if the initial segmentation of a tooth contains a majority of mispredicted facets, TSRNet may spread incorrect labels to the whole teeth, potentially resulting in fused teeth. Motivated by this, we introduce an interactive refinement scheme that allows users to click on the high-resolution original mesh and expand the influence area to the desired region, correcting it to the accurate label. We aim for users to effortlessly select the modification area with minimal interaction, while our algorithm provides an impact range that aligns with their expectations. This range should generally remain limited, as excessive expansion of the

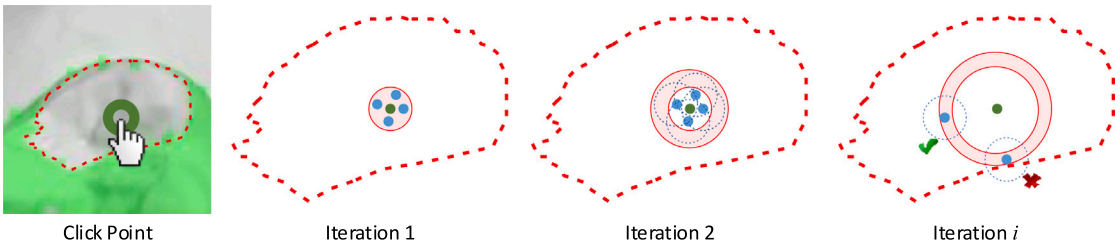


Fig. 5. Illustration of interactive refinement. The green point represents the click point, the red dashed line indicates the desired influence region, blue points mark additional diffusion points, the light red area shows the diffusion set $\mathbf{S}_d$, and the blue dashed line represents k-NN. The whole area within the larger red circle represents the selection set $\mathbf{S}_r^i$ of each iteration.

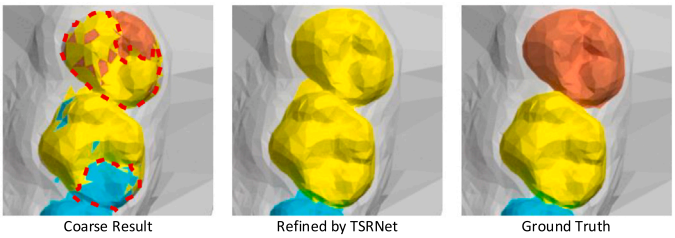


Fig. 6. Failure case of TSRNet. The red dashed line indicates the desired correction region.

modification area may lead to suboptimal results. We observed that the ideal range is typically enclosed by the initial segmentation boundary and the target segmentation boundary.

Specifically, we first apply k-nn ($k = 1$) to map the predicted results and boundary map from the simplified mesh back onto the original mesh. Given a user's click point p_c, we first choose its k-nearest neighbors ($k = 10{,}000$) to form a local patch, allowing for efficient processing on the local patch $\mathbf{S}_p$ within the high-resolution original mesh. The target boundary is defined as the union of boundary points $\mathbf{B}$ from both the initial segmentation and $\mathbf{B}_c$ the predicted boundary map $\mathbf{B}_m$: $\mathbf{B} = \mathbf{B}_c \cup \mathbf{B}_m$. We define the selection point set of the influence region as $\mathbf{S}_r$, and the set of influence region at iteration i as $\mathbf{S}_r^i$.

Initially, the click point p_c is included in the selection set $\mathbf{S}_r^0$. Then, its k-nearest neighbors ($k = 5$) are included in the $\mathbf{S}_r^1$ in the first iteration, with the difference recorded as the diffusion set $\mathbf{S}_d = \mathbf{S}_r^1 - \mathbf{S}_r^0$. The point set of the local patch is updated: $\mathbf{S}_p = \mathbf{S}_p - \mathbf{S}_r^1$. Similarly, in the subsequent iteration i, we calculate the k-nearest neighbors in $\mathbf{S}_p$ for each point in the diffusion set $\mathbf{S}_d$ and include them to the selection set $\mathbf{S}_r^i$, updating the diffusion set accordingly: $\mathbf{S}_d = \mathbf{S}_r^i - \mathbf{S}_r^{i-1}$. And update the patch set $\mathbf{S}_p = \mathbf{S}_p - \mathbf{S}_r^i$. Additionally, if some points in the updated diffusion set $\mathbf{S}_d$ is boundary points in $\mathbf{B}$, they are excluded from $\mathbf{S}_d$. The iteration will terminate if the maximum number of iterations ($I = 50$) is reached or if the selection set remains unchanged: $\mathbf{S}_r^i = \mathbf{S}_r^{i-1}$. The description of this process is shown in Fig. 5. Finally, all points in $\mathbf{S}_r$ are corrected to the user-specified label.

3.4. Boundary smoothing

Mapping labels from the simplified mesh back to the original mesh using k-nn can introduce slight inaccuracies along the boundaries, resulting in rough boundaries. To address this, we further smooth the tooth boundaries using the improved fuzzy clustering algorithm proposed by Xu et al. [26] and TeethGNN [17], which can be directly applied to high-resolution meshes. However, since Teeth3DS [25] provides vertex-wise labels, we modified the graph cuts algorithm adapted specifically for vertices to facilitate intuitive evaluation.

Similar to TeethGNN, we first construct a graph as the fuzzy boundary region for each tooth. The initial boundary vertices of this fuzzy region are defined as vertices with adjacent vertices carrying different labels. We then iteratively expand the fuzzy region, including each vertex and its adjacent vertices as nodes of the graph, and the edges of the mesh connecting vertices as the edges of the graph. In the final iteration, newly included tooth vertices form the source set $\mathcal{S}$, and gingiva vertices form the target set $\mathcal{T}$. Unlike facet-wise labels, vertex-wise labels lack clearly defined boundary edges. Even so, we still prioritize cutting edges at concave angles for vertices. Therefore, we follow the capacity calculation formula and parameters from TeethGNN to cut edges for vertices.

4. Experiment

4.1. Implementation details

4.1.1. Network details

We implement our method using PyTorch deep learning framework. As shown in Fig. 3, we utilize facet features $\mathbf{P} \in \mathbb{R}^{N\times 15}$ as the input for CBANet. The structure begins with a geometric feature extractor, which consists of a spatial transformer and five static EdgeConv modules that each output an $N \times 64$ feature map. The subsequent global embedding has a dimension of 1024. Both global and local features are concatenated to form the $N \times 1344$ point-wise features, which are then decoded to $\mathbf{F} \in \mathbb{R}^{N\times D}(D = 256)$. Throughout the network architecture, the k values are empirically set to 20. Furthermore, the number of candidates M is preset to 100.

4.1.2. Training setup

All experiments are conducted in an environment equipped with an Intel i7-11700K CPU, an NVIDIA GeForce RTX 3080Ti GPU, and 16 GB of RAM. The training process is set to run for a maximum of 200 epochs, utilizing an Adam optimizer with a weight decay parameter of 0.0001 to help prevent overfitting. The batch size for the training is set to 2. We employ a one-cycle learning rate scheduler to optimize the learning process. This scheduler starts with an initial learning rate determined by dividing the maximum learning rate of 0.001 by a division factor of 25, effectively setting the initial learning rate. We set the loss weights λ_b, λ_d, λ_e, λ_m, λ_c, and λ_s to 1.0, 5.0, 1.0, 10.0, 1.0 and 10.0, respectively. Notably, due to the initial insufficient accuracy of the boundary and geodesic distance maps to support the extraction of sampling points, we exclusively train the point-wise encoder for the first 20 epochs.

4.2. Experimental setup

4.2.1. Dataset

We conduct our 3D tooth segmentation experiments using the publicly available Teeth3DS [25] dataset. This benchmark comprises 1800 high-resolution 3D tooth models, divided into 900 maxillary (upper teeth) and 900 mandibular (lower teeth) models. All tooth models are annotated with FDI notation, resulting in 33 categories. Each mesh is labeled with 17 classes, including the gingiva and 16 individual teeth. We follow the official splits for each model type (maxillary or mandibular), allocating 600 models for training and the remaining 300 models for testing. To facilitate intuitive evaluation and comparison, we present the experimental results derived from simplified tooth meshes.

4.2.2. Metrics

We utilize three metrics from TSRNet [38] to evaluate the 3D tooth segmentation performance of CBANet, which include overall accuracy (OA), mean Intersection over Union for teeth (T-mIoU), and Intersection over Union for tooth boundaries (B-IoU). OA is defined as the proportion of correctly predicted facets among all facets:

$$OA = \frac{N_c}{N}. \tag{10}$$

Here, N_c represents the number of correctly predicted facets, while N denotes the total number of facets.

Given that gingiva facets constitute the majority of facets in 3D tooth models, the mIoU for all categories can obscure the intuitive representation of tooth instance segmentation performance. Therefore, we calculate the average IoU between the predicted labels and the ground truth for all tooth instances within each tooth model, which is formulated as follows:

$$T\text{-}mIoU = \frac{1}{T}(\sum_{i=1}^{T} \frac{P_i \cap G_i}{P_i \cup G_i}), \tag{11}$$

where T represents the number of tooth instances in a tooth model. P_i denotes the set of facets predicted as the ith tooth, while G_i corresponds to the set of ground truth facets.

In addition, the accuracy of the tooth boundaries is a crucial aspect of performance evaluation. To intuitively verify the alignment between the predicted boundaries B_p and the ground truth boundaries B_g, we introduce the boundary IoU, which can be calculated as follows:

$$B\text{-}IoU = \frac{B_p \cap B_g}{B_p \cup B_g}. \tag{12}$$

We consider a vertex to be a boundary vertex if its adjacent faces are not labeled identically in a mesh.

4.2.3. Competing methods

We conduct a comparative analysis of our method against seven advanced algorithms, which include three point cloud segmentation algorithms (DGCNN [8], and Point Transformer [40], ISBNet [19]) and five state-of-the-art 3D tooth segmentation algorithms (TSegNet [21], MeshSegNet [3], iMeshSegNet [4], TSGCNet [10], and TeethGNN [17]). Moreover, We further compare the results of TeethGNN [17] after optimization with TSRNet [38] (TG+TSR).

4.3. Comparison on Teeth3DS

We report the overall quantitative comparison results on the Teeth3DS dataset, as shown in Table 2. CBANet achieves the best performance in the 3D tooth segmentation across the evaluation metrics OA, T-mIoU, and B-IoU, significantly outperforming competing methods. Consistent with this, Fig. 7 presents visualization examples from various algorithms, including two upper tooth models, two lower tooth models, and a tooth model with a lateral view. We observe that point-based segmentation methods inevitably produce isolated prediction errors and imprecise tooth boundaries. While ISBNet [19] outperforms other general segmentation methods, it exhibits suboptimal performance compared to CBANet when applied to tooth data. This is mainly attributed to the limitations of its instance-aware farthest point sampling strategy, which is not well-suited for closely arranged teeth on jaw models. TeethGNN [17] relies heavily on prior knowledge and clustering. It cannot guarantee the overall optimal clustering hyperparameter settings are adapted to every tooth model in the dataset. This limitation becomes particularly pronounced in some challenging cases, such as closely spaced teeth or crowded teeth, which often result in incorrect clustering and the fusion of teeth. Even with TSRNet [38] optimization (TG+TSR), these issues of semantic errors for entire teeth cannot be resolved.

In contrast, our CBANet showcases remarkable effectiveness and efficiency in 3D tooth segmentation, performing exceptionally well in these challenging cases. This enhancement primarily stems from our strategic employment of geodesic distance maps in the selection of sampling points for candidate feature generation. Geodesic distance maps are highly sensitive to small or missing teeth, ensuring that the center-aware farthest point sampling strategy achieves comprehensive and uniform coverage across all teeth. This effectively prevents tooth omission and mislocalization. Furthermore, encoding high-quality instance features from the sampled points allows CBANet to accurately highlight specific tooth regions. When combined with the boundary constraints provided by the boundary-aware mask features, this effectively mitigates issues of over-segmentation and fusion between adjacent teeth. The integration of auxiliary boundary maps also sharpens the definition of tooth boundaries, enabling CBANet to deliver superior results at tooth boundaries. As a result, CBANet consistently outperforms previous methods in these challenging cases and achieves smoother tooth boundaries, demonstrating superior generalization capabilities in accommodating tooth models of varied shapes, as shown in Fig. 7.

Additionally, we select 20 cases with imperfect initial prediction results, including 10 models from the upper jaw and 10 from the lower jaw, to evaluate the performance of our interactive refinement scheme. We first map the results from the simplified meshes onto the high-resolution original models, then correct the erroneous patches on the original mesh through user clicks. Since the mapping process can create rough boundaries, we further smooth the tooth boundaries with the graph cuts algorithm. The visualization examples are provided in Fig. 8. The interactive refinement (IR) effectively corrects large areas of prediction errors in the initial results, a task that is challenging for graph cuts (GC) alone. We further smooth the refined results to achieve optimal outcomes. Our results demonstrate that interactive refinement significantly corrects mispredicted patches on both gingiva and teeth. Even for missed, fused, or semantically incorrect teeth, this method effectively restores the labels across the entire tooth region. The quantitative results on the original meshes are shown in Table 3 using OA, T-mIoU, and B-IoU metrics. Unlike the boundary point definition in simplified meshes, we consider a point to be a boundary point if there are points with different labels within its k-neighborhood ($k = 10$).

4.4. Computation time

The inference time of CBANet on a simplified mesh with 10,000 facets is 96 ms, which includes 71 ms for network inference and 25 ms for the NMS computation on 100 candidates. The computation time for our interactive refinement scheme mainly depends on the area of the patches being corrected. Larger erroneous patches require more iterations, and as the iterations progress, the number of points in the diffusion set $\mathbf{S}_d$ increases, further extending computation time. Typically, patches requiring 10 iterations take around 0.6 s, while those needing 20 iterations take approximately 2.8 s. For 30 iterations, the time extends to about 5.7 s, and for 40 iterations, it reaches 9.2 s. With the maximum limit of 50 iterations, the process may take up to 15.4 s, which is generally acceptable for practical applications. Additionally, smoothing the boundaries for a jaw model takes approximately 3.9 s, while processing an individual tooth takes an average of 0.27 s.

5. Ablation study

We conduct a series of ablation studies to validate the segmentation performance of CBANet on 3D tooth models and to thoroughly analyze the impact of various components and hyperparameters within the network architecture. All experiments are evaluated using three key metrics: OA, T-mIoU, and B-IoU.

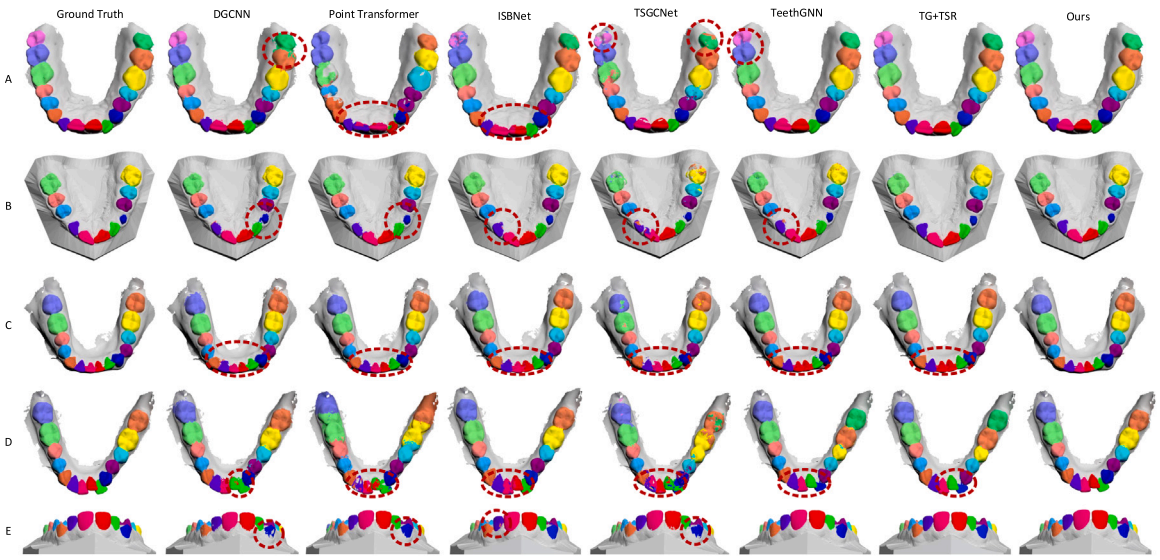


Fig. 7. Visualization of 3D tooth segmentation results from different methods, including five tooth models: two upper jaw models, two lower jaw models, and a lateral view of a tooth model. A: An upper jaw model with a complete set of 16 teeth. B: An upper jaw model with missing teeth and small-scale teeth. C: A lower jaw model with attachments. D: A lower jaw model with crowded teeth. E: A lateral view of a tooth model with ectopic teeth. Red dashed circles highlight some challenging regions where our method significantly excels in competing methods.

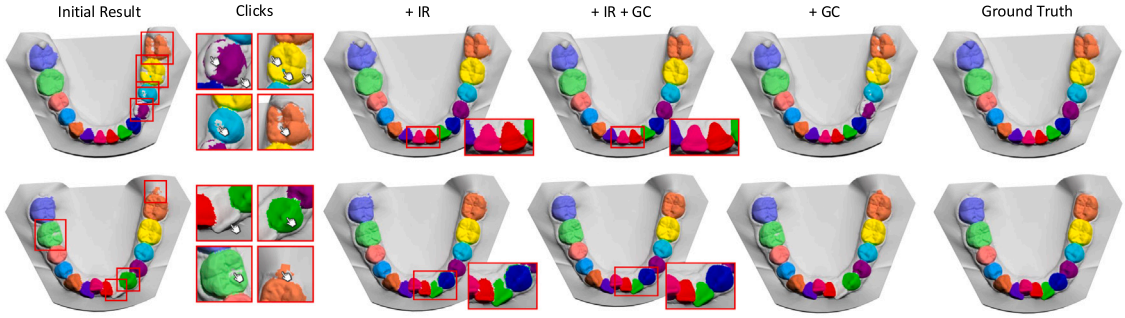


Fig. 8. Visualization of 3D tooth segmentation results on the original mesh. We apply interactive refinement (IR) to correct mispredicted patches of initial results, followed by graph cuts (GC) to further smooth the tooth boundaries.

Table 2
Quantitative comparisons of the 3D tooth model segmentation results with various methods, evaluated in terms of overall accuracy (OA), teeth mIoU (T-mIoU), and boundary IoU (B-IoU).

	Upper			Lower			All		
	OA (%)↑	T-mIoU (%)↑	B-IoU (%)↑	OA (%)↑	T-mIoU (%)↑	B-IoU (%)↑	OA (%)↑	T-mIoU (%)↑	B-IoU (%)↑
PointTransformer [40]	89.71	76.43	41.70	85.28	69.18	35.83	87.49	72.80	38.76
DGCNN [8]	92.94	83.25	41.54	89.26	78.43	35.35	91.10	80.84	38.44
ISBNet [19]	94.11	84.35	52.93	91.14	79.42	45.93	92.62	81.88	49.43
TSegNet [21]	85.96	69.22	43.83	73.88	53.42	27.63	79.92	61.32	35.73
MeshSegNet [3]	83.46	69.75	31.82	74.19	58.77	24.33	78.82	64.26	28.07
iMeshSegNet [4]	86.67	71.45	32.59	81.51	67.05	28.64	84.09	69.25	30.61
TSGCNet [10]	92.59	82.98	45.60	90.27	82.01	46.64	91.43	82.49	46.12
TeethGNN [17]	95.18	90.00	69.32	90.20	84.46	59.08	92.69	87.23	64.20
TG [17]+TSR [38]	95.52	90.95	74.29	91.07	85.99	**66.62**	93.29	88.47	**70.45**
CBANet	**96.59**	**92.65**	**74.58**	**92.88**	**88.64**	64.14	**94.73**	**90.64**	69.36

5.1. Impacts of different candidate amounts

The preset number of candidates is a critical hyperparameter in our method, and we analyze CBANet's segmentation performance across various candidate counts. If the sampling points are too few, center-aware FPS may fail to extract all tooth centers from the predicted geodesic distance map. Furthermore, the second-stage FPS cannot ensure a complete recall of sampling points for all tooth instances. Increasing the number of sampling points not only guarantees comprehensive coverage of tooth instances but also enhances the likelihood of selecting high-quality candidates. However, excessively high candidate counts increase the computational burden on the neural network. As

Table 3
Quantitative results on the original meshes, including the results of the initial segmentation, after interactive refinement (+IR), after boundary smoothing (+GC), and with both interactive refinement and subsequent boundary smoothing (+IR+GC).

	Upper			Lower		
	OA	T-mIoU	B-IoU	OA	T-mIoU	B-IoU
Initial Result	93.96	87.40	44.12	90.94	87.56	45.85
+IR	96.06	91.63	45.59	94.80	90.86	48.18
+GC	95.28	92.70	60.95	92.58	90.67	62.55
+IR+GC	**97.41**	**94.95**	**62.64**	**96.47**	**94.28**	**65.49**

Table 4
Quantitative results with different candidate amounts M.

M	Upper			Lower		
	OA	T-mIoU	B-IoU	OA	T-mIoU	B-IoU
20	94.57	87.38	70.10	89.56	80.68	61.27
40	96.30	91.92	73.22	92.23	86.40	64.26
60	96.29	91.94	73.79	92.82	88.53	**65.12**
80	96.25	92.05	73.42	92.49	87.05	64.35
100	**96.59**	**92.65**	**74.58**	**92.88**	**88.64**	64.14

Table 5
Results of the ablation study on feature aggregation.

PW-GCN	CW-SA	Upper			Lower		
		OA	T-mIoU	B-IoU	OA	T-mIoU	B-IoU
		95.88	91.07	73.23	92.35	86.74	64.34
✓		96.35	91.41	74.49	92.58	87.65	**64.89**
✓	✓	**96.59**	**92.65**	**74.58**	**92.88**	**88.64**	64.14

Table 6
Results of the ablation study on boundary map (bmap).

	Upper			Lower		
	OA	T-mIoU	B-IoU	OA	T-mIoU	B-IoU
w/o bmap	95.52	91.01	71.45	91.63	86.11	61.52
w/ bmap	**96.59**	**92.65**	**74.58**	**92.88**	**88.64**	**64.14**

demonstrated in Table 4, with $M = 20$, the segmentation results are markedly poor. When M exceeds 40, its impact on CBANet's performance becomes negligible.

5.2. Effectiveness of feature aggregation

During the encoding process of candidate features, we integrate point-wise graph convolution and candidate-wise self-attention blocks within the feature aggregation. The point-wise graph convolution block is aimed at aggregating homogeneous features from the local neighborhood context of sampling points, transforming them into representative candidate-wise features. The candidate-wise self-attention block forces each candidate feature to facilitate information interaction with candidates. To validate the rationality and effectiveness of this design, we conduct a quantitative analysis of these two blocks, with the results shown in Table 5. We observe that both point-wise (PW-GCN) and candidate-wise (CW-SA) blocks significantly enhance segmentation performance. Compared to using sampled point features directly, this improvement more effectively generates representative instance representations that highlight tooth regions.

5.3. Effects of boundary map

In our method, the boundary map primarily serves to limit the influence range of click points during the interactive refinement process, defining the diffusion endpoint. Additionally, using the boundary map as auxiliary supervision can impose boundary constraints on sampled candidate points, thereby generating masks that sharpen the boundary features. To assess its impact, we evaluate CBANet's prediction results with (w/ bmap) and without (w/o bmap) the boundary map. As shown in Table 6, incorporating the boundary map as auxiliary supervision enhances CBANet's segmentation performance.

Table 7
Quantitative results with different k values in CBANet.

k	Upper			Lower		
	OA	T-mIoU	B-IoU	OA	T-mIoU	B-IoU
5	95.82	90.81	67.52	91.27	85.87	56.97
10	96.32	92.16	72.17	91.68	86.40	61.51
15	96.35	92.25	74.41	92.65	88.04	**64.48**
20	**96.59**	**92.65**	**74.58**	**92.88**	**88.64**	64.14

Table 8
Quantitative results with different lower thresholds.

ϵ	Upper			Lower		
	OA	T-mIoU	B-IoU	OA	T-mIoU	B-IoU
0	96.33	92.23	73.38	91.48	85.75	62.82
0.1	96.42	92.36	73.47	92.50	87.11	64.03
0.2	**96.59**	**92.65**	**74.58**	**92.88**	**88.64**	**64.14**
0.3	96.42	92.30	73.32	92.20	86.68	63.78

5.4. Impacts of different k-value in CBANet

In CBANet, both the static EdgeConv for geometric feature extraction and the point-wise graph convolution in feature aggregation rely on k-NN, with k consistently set to a default of 20. The k-value defines the receptive field for graph convolution. Larger k-values increase the receptive field but also introduce computational burden. To evaluate this, we examine CBANet's segmentation performance with different k-values, as shown in Table 7. Results indicate that on simplified meshes with 10,000 facets, a smaller k-value yields suboptimal results, while increasing k beyond 15 has a negligible impact on accuracy.

5.5. Impacts of different lower threshold ϵ_l

In our Center-Aware Farthest Point Sampling (CA-FPS) strategy, we introduce a lower threshold ϵ_l based on the predicted distance map to select a reliable set of tooth points. This design serves two key purposes: first, to ensure that sampled points are located within tooth regions and are sufficiently distant from the tooth boundaries, and second, to prevent cross-boundary feature aggregation by ensuring that the local neighborhood used for encoding instance features belongs to the same tooth. This promotes the generation of high-quality candidate instance features. We conduct an analysis on the impact of different lower threshold values, as illustrated in Table 8. While our NMS strategy is effective at selecting high-quality tooth instance masks and provides strong fault tolerance, we observe a noticeable drop in segmentation performance for lower teeth when the threshold is set to 0. However, when the threshold exceeds 0.1, its influence on the final segmentation results becomes minimal. These results further validate the effectiveness and robustness of our CA-FPS strategy.

6. Conclusion

We introduce CBANet, a 3D tooth instance segmentation network, alongside an interactive refinement strategy. CBANet learns center- and boundary-aware features via dual branches supervised by boundary and geodesic distance maps. Based on the distance map, we design a center-aware farthest point sampling strategy to ensure comprehensive and uniformly distributed candidate points. The feature aggregation module encodes local homogeneous point features and inter-point relations to form representative candidate instance features. The prediction heads then combine these candidate and mask features to predict the candidate masks, as well as the corresponding class probabilities and IoU

scores. Leveraging the well-designed NMS, CBANet efficiently merges high-quality masks to deliver precise segmentation results. Our interactive refinement scheme takes user-clicked points and the predicted boundary map as inputs, and progressively expands the influence area through iterative diffusion to accurately select and correct the target patch with the specified label.

CRediT authorship contribution statement

Hairong Jin: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Jianwen Lou:** Writing – review & editing. **Zhiguo Lu:** Validation. **Teng Wu:** Validation. **Kun Zhou:** Funding acquisition. **Youyi Zheng:** Funding acquisition, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

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

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 62172363).

Data availability

The code will be made publicly available.

References

[1] Jang TJ, Kim KC, Cho HC, Seo JK. A fully automated method for 3D individual tooth identification and segmentation in dental CBCT. IEEE Trans Pattern Anal Mach Intell 2021;44(10):6562–8.
[2] Rodrigues MAF, Silva WB, Neto MEB, Gillies DF, Ribeiro IM. An interactive simulation system for training and treatment planning in orthodontics. Comput Graph 2007;31(5):688–97.
[3] Lian C, Wang L, Wu T-H, Wang F, Yap P-T, Ko C-C, Shen D. Deep multi-scale mesh feature learning for automated labeling of raw dental surfaces from 3D intraoral scanners. IEEE Trans Med Imaging 2020;39(7):2440–50.
[4] Wu T-H, Lian C, Lee S, Pastewait M, Piers C, Liu J, Wang F, Wang L, Chiu C-Y, Wang W, et al. Two-stage mesh deep learning for automated tooth segmentation and landmark localization on 3D intraoral scans. IEEE Trans Med Imaging 2022;41(11):3158–66.
[5] Wu K, Chen L, Li J, Zhou Y. Tooth segmentation on dental meshes using morphologic skeleton. Comput Graph 2014;38:199–211.
[6] Qi CR, Su H, Mo K, Guibas LJ. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 652–60.
[7] Qi CR, Yi L, Su H, Guibas LJ. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Adv Neural Inf Process Syst 2017;30.
[8] Wang Y, Sun Y, Liu Z, Sarma SE, Bronstein MM, Solomon JM. Dynamic graph cnn for learning on point clouds. Acm Trans Graph (TOG) 2019;38(5):1–12.
[9] Xiong H, Li K, Tan K, Feng Y, Zhou JT, Hao J, Liu Z. TFormer: 3D tooth segmentation in mesh scans with geometry guided transformer. 2022, arXiv preprint arXiv:2210.16627.
[10] Zhang L, Zhao Y, Meng D, Cui Z, Gao C, Gao X, Lian C, Shen D. TSGCNet: Discriminative geometric feature learning with two-stream graph convolutional network for 3D dental model segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, p. 6699–708.
[11] Tang L, Zhan Y, Chen Z, Yu B, Tao D. Contrastive boundary learning for point cloud segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 8489–99.
[12] Liu Y, Li W, Liu J, Chen H, Yuan Y. GRAB-net: Graph-based boundary-aware network for medical point cloud segmentation. IEEE Trans Med Imaging 2023.
[13] Yang B, Wang J, Clark R, Hu Q, Wang S, Markham A, Trigoni N. Learning object bounding boxes for 3d instance segmentation on point clouds. Adv Neural Inf Process Syst 2019;32.
[14] Yi L, Zhao W, Wang H, Sung M, Guibas LJ. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 3947–56.
[15] Jiang L, Zhao H, Shi S, Liu S, Fu C-W, Jia J. Pointgroup: Dual-set point grouping for 3d instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, p. 4867–76.
[16] Vu T, Kim K, Luu TM, Nguyen T, Yoo CD. Softgroup for 3d instance segmentation on point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 2708–17.
[17] Zheng Y, Chen B, Shen Y, Shen K. Teethgnn: Semantic 3d teeth segmentation with graph neural networks. IEEE Trans Vis Comput Graphics 2022.
[18] He T, Yin W, Shen C, Van den Hengel A. Pointinst3d: Segmenting 3d instances by points. In: European conference on computer vision. Springer; 2022, p. 286–302.
[19] Ngo TD, Hua B-S, Nguyen K. ISBNet: a 3D point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 13550–9.
[20] Zhuang S, Wei G, Cui Z, Zhou Y. Robust hybrid learning for automatic teeth segmentation and labeling on 3D dental models. IEEE Trans Multimed 2023.
[21] Cui Z, Li C, Chen N, Wei G, Chen R, Zhou Y, Shen D, Wang W. TSegNet: An efficient and accurate tooth segmentation network on 3D dental model. Med Image Anal 2021;69:101949.
[22] Zanjani FG, Moin DA, Claessen F, Cherici T, Parinussa S, Pourtaherian A, Zinger S, de With PH. Mask-MCNet: Instance segmentation in 3D point cloud of intra-oral scans. In: Medical image computing and computer assisted intervention–MICCAI 2019: 22nd international conference, Shenzhen, China, October 13–17, 2019, proceedings, part v 22. Springer; 2019, p. 128–36.
[23] Qiu L, Ye C, Chen P, Liu Y, Han X, Cui S. Darch: Dental arch prior-assisted 3d tooth instance segmentation with weak annotations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 20752–61.
[24] Dascalu T, Ibragimov B. Assignment theory-augmented neural network for dental arch labeling. In: International conference on medical image computing and computer-assisted intervention. Springer; 2023, p. 295–304.
[25] Ben-Hamadou A, Smaoui O, Chaabouni-Chouayakh H, Rekik A, Pujades S, Boyer E, Strippoli J, Thollot A, Setbon H, Trosset C, et al. Teeth3ds: a benchmark for teeth segmentation and labeling from intra-oral 3d scans. 2022, arXiv preprint arXiv:2210.06094.
[26] Xu X, Liu C, Zheng Y. 3D tooth segmentation and labeling using deep convolutional neural networks. IEEE Trans Vis Comput Graphics 2018;25(7):2336–48.
[27] He K, Gkioxari G, Dollár P, Girshick R. Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. 2017, p. 2961–9.
[28] Liu S-H, Yu S-Y, Wu S-C, Chen H-T, Liu T-L. Learning gaussian instance segmentation in point clouds. 2020, arXiv preprint arXiv:2007.09860.
[29] Zhao W, Yan Y, Yang C, Ye J, Yang X, Huang K. Divide and conquer: 3d point cloud instance segmentation with point-wise binarization. In: Proceedings of the IEEE/CVF international conference on computer vision. 2023, p. 562–71.
[30] Wu Y, Shi M, Du S, Lu H, Cao Z, Zhong W. 3D instances as 1d kernels. In: European conference on computer vision. Springer; 2022, p. 235–52.
[31] Ji Z, Liu L, Chen Z, Wang G. Easy mesh cutting. In: Computer graphics forum. Vol. 25, Wiley Online Library; 2006, p. 283–91.
[32] Zhao M, Ma L, Tan W, Nie D. Interactive tooth segmentation of dental models. In: 2005 IEEE engineering in medicine and biology 27th annual conference. IEEE; 2006, p. 654–7.
[33] Zou B-j, Liu S-j, Liao S-h, Ding X, Liang Y. Interactive tooth partition of dental mesh base on tooth-target harmonic field. Comput Biol Med 2015;56:132–44.
[34] Jiang X, Xu B, Wei M, Wu K, Yang S, Qian L, Liu N, Peng Q. C2F-3DToothSeg: Coarse-to-fine 3D tooth segmentation via intuitive single clicks. Comput Graph 2022;102:601–9.
[35] Chen X, Zhao Z, Yu F, Zhang Y, Duan M. Conditional diffusion for interactive segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. 2021, p. 7345–54.
[36] Xu L, Li S, Chen Y, Chen J, Huang R, Wu F. ClickAttention: Click region similarity guided interactive segmentation. 2024, arXiv preprint arXiv:2408.06021.
[37] Kontogianni T, Celikkan E, Tang S, Schindler K. Interactive object segmentation in 3d point clouds. In: 2023 IEEE international conference on robotics and automation. ICRA, IEEE; 2023, p. 2891–7.
[38] Jin H, Shen Y, Lou J, Zhou K, Zheng Y. TSRNet: A dual-stream network for refining 3D tooth segmentation. IEEE Trans Vis Comput Graphics 2024.
[39] Garland M, Heckbert PS. Surface simplification using quadric error metrics. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques. 1997, p. 209–16.
[40] Zhao H, Jiang L, Jia J, Torr PH, Koltun V. Point transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. 2021, p. 16259–68.