LETA: Tooth Alignment Prediction Based on Dual-branch Latent Encoding

Zefeng Shi, Zijie Meng, Ruizhe Chen, Yang Feng, Zeyu Zhao, Jin Hao, Bing Fang, Zuozhu Liu, *Member, IEEE*, and Youyi Zheng

***Abstract*—Accurately determining the clinical positions for each tooth is essential in orthodontics, while most existing solutions heavily rely on inefficient manual design. In this paper, we present the LETA, a dual-branch Latent Encoding based 3D Tooth Alignment. Our system takes as input the segmented individual 3D tooth meshes in the Intra-oral Scanner (IOS) dental surfaces, and automatically predicts the proper 3D pose transformation for each tooth. LETA includes three components: an Encoder that learns a latent code of dental pointcloud, a Projector that transforms the latent code of misaligned teeth to predicted aligned ones, and a Solver to estimate the transformation between different dental latent codes. A key novelty of LETA is that we extract the features from the ground truth (GT) aligned teeth to guide network learning during training. To effectively learn tooth features, our Encoder employs an improved point-wise convolutional operation and an attention-based network to extract local shape features and global context features respectively. Extensive experimental results on a large-scale dataset with 9,868 IOS surfaces demonstrate that LETA can achieve state-of-the-art performance. A further clinical applicability study reveals that our method can reduce orthodontists' workload over 60% compared to starting tooth alignment from scratch, demonstrating the strong potential of deep learning for future digital dentistry.**

***Index Terms*—Orthodontics, tooth alignment, attention, pointcloud.**

Manuscript received 18 February 2023; revised 3 April 2024; accepted 3 June 2024. Date of publication 20 June 2024; date of current version 1 August 2025. This work was supported by the National Natural Science Foundation of China under Grant 62106222 and Grant 62172363, in part by the Natural Science Foundation of Zhejiang Province, China under Grant LZ23F020008, and in part by the Zhejiang University-Angelalign Technology Inc. R&D Center for Intelligent Healthcare. Recommended for acceptance by Q. Huang. *(Zefeng Shi, Zijie Meng, and Ruizhe Chen contributed equally to this work.) (Corresponding authors: Bing Fang, Zuozhu Liu; Youyi Zheng.)*

Zefeng Shi and Youyi Zheng are with the State Key Lab of CAD&CG, Zhejiang University, Zhejiang 310058, China (e-mail: shiuangustc@gmail.com; youyizheng@zju.edu.cn).

Zijie Meng, Ruizhe Chen, and Zuozhu Liu are with the Zhejiang University-University of Illinois at Urbana Champaign Institute, Zhejiang University, Zhejiang 314400, China (e-mail: zijie.22@intl.zju.edu.cn; ruizhec.21@intl.zju.edu.cn; zuozhuliu@intl.zju.edu.cn).

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

Zeyu Zhao is with ChohoTech Company Ltd., Hangzhou 310000, China (e-mail: zzy8200@gmail.com).

Jin Hao is with the Harvard School of Dental Medicine, Harvard University, Boston, MA 02115 USA (e-mail: jinhaoscrb@gmail.com).

Bing Fang is with the Department of Orthodontics, Shanghai Ninth People's Hospital, College of Stomatology, Shanghai Jiao Tong University School of Medicine, Shanghai 200011, China (e-mail: braces_dr@hotmail.com).

Digital Object Identifier 10.1109/TVCG.2024.3413857

I. Introduction

RECENT research has witnessed the emerging success of artificial intelligence and deep learning techniques for digital dentistry, such as 3D tooth semantic segmentation in IOS dental surfaces [1], [2], [3], [4], [5], 2D tooth and jawbone segmentation in Cone-Beam CT (CBCT) images[6], [7], [8], [9], automatic tooth alignment[10], [11], [12], [13], etc. Such techniques are highly demanded for many dental applications, e.g. orthodontics and implant. Alongside the treatment process of orthodontics, an essential step is to come up with appropriate and clinically applicable tooth alignment plans, which will determine the final orientation and position of each tooth to bring patients beautiful aligned teeth. More specifically, with an input set of misaligned teeth from a patient, the goal of tooth alignment is to predict a rigid transformation for each tooth to help align all teeth neatly while meeting a set of orthodontic rules.

Nowadays, the conflict between the rapid growth of patients' need for orthodontic treatment and the limited number of dentists makes it even harder to offer satisfactory treatment planning for all patients in a reasonably short time [14], [15], [16], [17]. A satisfactory alignment should balance both medical and aesthetical perspectives. It requires an orthodontist to make several proposals and find the optimal solution. Besides, professional orthodontists need to perform a large number of manual operations in the orthodontic software to get orthodontically aligned teeth. This is a cumbersome and time-consuming process. In addition, the performance of manual tooth alignment mainly depends on the experience and skills of individual orthodontists, which may be influenced by personal biases. Therefore, an efficient and high-quality automatic tooth alignment system is of high necessity to assist dentists for more efficient and clinically applicable treatment planning.

Automatic tooth alignment remains a challenging task, while effective and clinically applicable methods are yet to be developed. The key challenges are as follows. First, while automatic tooth alignment methods try to formulate the tooth alignment as a mathematical problem, e.g. regressing the corresponding rotation and translation of each tooth by regarding them as rigid bodies, its connection to clinical rules, such as "Andrew's Six Keys" [18] which identify the necessary conditions for proper tooth alignment, need to be established either explicitly or implicitly. Simply defining a regression loss over the rigid transformations between the misaligned and aligned teeth might not lead to clinically applicable solutions. Second, the tooth alignment tasks are very complicated, while an automatic solution needs to deal

1077-2626 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

with diversified patient samples. Lots of clinical features, such as the number, shape and size of teeth might vary significantly among patients. For example, some patients might only have less than 28 permanent teeth, while others might have 32 permanent teeth, not to mention the even more complicated morphological or anatomical features, such as different dental arch shapes, malocclusion types, etc. Such diversified inputs impose great challenges for an automatic solution with good generalization ability. Finally, there is no public dataset, no matter small or large-scale, for tooth alignment due to privacy issues, while the effectiveness and clinical applicability need to be evaluated with a relatively large dataset for reliable solutions.

Several attempts have been launched for automatic or semi-automatic tooth alignment. In the early stage, geometry-based tooth alignment methods are proposed to automatically make alignment predictions. These methods require an in-depth understanding of orthodontics. For example, Kumar et al. [19] have presented a method to establish dental occlusion via a simulation-based approach. This approach models the alignment process as a spring-mass system and uses a set of constraints obtained from the experience of professional orthodontists to connect teeth. Relying on specific rules in constrained circumstances, these methods do not generalize well in many cases which lose tooth features due to tooth defects, missing teeth, and dental diseases. With the rapid development of machine learning algorithms, several deep learning based tooth alignment methods are proposed. These methods demonstrate great superiority in good generalization, which can learn orthodontic rules from clinical data and have been proven to be more effective. TANet[11] and TAligNet[12] consider tooth alignment as a regression problem. These methods take misaligned teeth as input and employ deep models to regress the transformations between the input and the aligned teeth. However, the neural networks in such methods predominantly account for the geometric information in the misaligned teeth while ignoring the essential geometric information and implicit clinical rules derived from the aligned teeth, which could merit effective guides for deep models during the training phase.

In this paper, we propose a dual-branch Latent Encoding based 3D Tooth Alignment, denoted as LETA. Different from previous deep learning based methods, our method exploits the information in GT aligned teeth during training for effective feature extraction and latent code alignment. In particular, the network consists of three main components: an Encoder for both shape and context feature extraction from misaligned and GT aligned teeth via an improved PointConv[20] and a self-attention based architecture [21], a Projector for aligning latent code from misaligned teeth to predicted aligned teeth, and a Solver for predicting per-tooth rotation and translation between two sets of teeth based on the difference of their latent codes.

Our method is based on a newly collected large-scale dataset with 9,868 pairs of misaligned and GT aligned 3D IOS tooth surfaces. We uniformly sample points on each individual tooth. Experiments show that our method can achieve average rotation error, translation error, and point-wise distance of 5.28°, 0.91mm, and 0.97mm, respectively, which significantly outperforms previous state-of-the-arts both qualitatively and quantitatively. Moreover, a clinical applicability study with several professional orthodontists on 50 additional real-world cases reveals that predictions generated by our method can surpass or approach qualified levels with respect to four scoring statistics (adjacent relationship, occlusion, tooth rotation, molar relationship), and reduce orthodontists' workload over 60%. To summarize, the main contributions of this work are as follows:

- We introduce a novel framework for 3D tooth alignment prediction based on a dual-branch latent encoding, which effectively exploits the geometric information and clinical rules in the GT aligned teeth to guide network training.
- We introduce a novel Encoder comprised of improved PointConv operation and self-attention mechanism to effectively extract both local shape and global context features of teeth.
- We demonstrate the effectiveness of our method with a clinical applicability study and experiments on a large-scale real-world dataset consisting of 9,868 pairs of misaligned and GT aligned 3D IOS dental surfaces, with significantly better performance than previous deep learning based methods.

II. Related Work

A. Automatic Tooth Alignment

Existing methods for tooth alignment can be categorized into rule-based geometry methods and data-driven deep learning methods. Rule-based methods [19], [22], [23], [24], [25] need to define related constraints (which heavily rely on the understanding of orthodontics) based on teeth features and iteratively optimize them. These rule-based methods are designed for specific circumstances, which have poor generalization ability and cannot be applied to complicated clinical tooth alignment. Deep learning based methods[11], [12] learn the rules of tooth alignment from the existing large amount of clinical data. Specifically, TAligNet[12] is a lightweight network for tooth alignment, which uses PointNet[26] for shape feature extraction and uses multi-layer convolution and multi-layer fully connected layers for target tooth position prediction. TANet[11] also uses PointNet to extract shape features, including the single tooth shape and the entire dentition shape. It subsequently defines a graph based on the spatial relationships among the teeth, and uses Gate Recurrent Unit (GRU) to extract context features based on the graph and the shape feature of each tooth. These methods both use PointNet for shape feature extraction. However, PointNet is not a very powerful model for extracting shape features in local regions. Furthermore, they only take misaligned teeth as input, and discard a lot of useful information in the GT aligned teeth, which could merit effective rules for tooth alignment. In contrast, our method based on dual-branch latent encoding effectively makes use of the GT aligned teeth data during training. Our experimental results show that such exploitation could lead to a more faithful learning of the mapping from misaligned teeth to predicted aligned ones.

B. Deep Learning for Pointcloud

Pointcloud is one of the commonly used representations of 3D data [27], [28]. Compared to images, pointcloud is unordered, sparse, and permutation-invariant so that existing deep models

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

such as CNN are inapplicable. To address this problem, PointNet [26] adopts a symmetric function, i.e. a max-pooling layer to achieve permutation-invariant. On the basis of PointNet, PointNet++[29] further introduces a hierarchical structure, which uses PointNet to capture local structures from the neighbors of each point. It also uses sampling layers to down-sample the pointcloud. Afterward, various attempts have been made for the effective aggregation of features of neighbors. These methods can be roughly categorized into continuous convolution methods[20], [30], [31], discrete convolution methods[32], [33], [34], and graph-based methods[35]. Continuous convolution methods map the relative positions of points in the neighborhood to the kernel of convolution, while discrete convolution methods divide the neighborhood space into discrete regions and use shared training parameters for points in the same region. Hence using multiple continuous convolution layers to extract features without sampling layers, the neighborhood of one point remains the same. Graph-based methods, e.g. DGCNN[35], further define explicit local graphs based on the proximity of latent representations. With a k-nearest neighbor method, DGCNN can learn local features in bottom Edge-conv blocks and global semantic features in top Edge-conv blocks. However, our IOS dental data are more complicated and of high resolution, which makes these frameworks perform unsatisfactorily.

C. Attention-Based Methods

Although CNN-based methods have achieved high performance in segmentation tasks, these approaches still have intrinsic limitations to learn global context information and long-range dependencies. Since the Transformer [21] was proposed, the attention mechanism has attracted widespread attention. The methods, such as BERT [36], have gradually replaced the Recurrent Neural Network (RNN) methods in Natural Language Processing (NLP). Inspired by its great success in NLP, Transformer has been brought into the vision domain [37], [38], [39]. In computer vision, Non-Local Neural Networks[40] use an attention mechanism to achieve the extraction of global features of images. Recently, Vision transformer (ViT)[41] is proposed to perform image classification in an NLP way. It separates the 2D image into patches and linearly aligns them with positional embeddings. These patches are input into a BERT-like transformer. Based on pre-training on a large dataset, ViT achieves comparable performance in the vision domain. Recently, Swin Transformer[42] is proposed, which makes use of window-based multi-head self-attention (W-MSA) and shifted window-based multi-head self-attention (SW-MSA) mechanism to build hierarchical attention. It achieves state-of-the-art performance on image classification, object detection and semantic segmentation tasks. The attention mechanisms have been used in pointcloud and graph deep learning as well [43], [44]. Pointcloud Transformer (PCT) [44] ingeniously proposes a deep learning framework for pointcloud processing. The main idea of PCT is to combine the inherent permutation invariance of pointcloud with the permutation invariant attention mechanism. In addition, PCT designs a coordinates-based input embedding module, offset attention module and adjacent embedding module to make Transformer more suitable for pointcloud. Inspired by the incredible performance of visual Transformers, in our method, we also employ the Transformer-based block to extract global context features of teeth and capture the spatial relationship among teeth.

III. Method

A. Overview

As illustrated in Fig. 1, there are three main components in our LETA: an Encoder to embed pointcloud to latent code via local shape and global context feature extraction, a Projector to transform the latent code of misaligned teeth to predicted aligned ones, and a Solver to infer the rigid pose transformation from the difference of latent codes. The details of these three components are described in the subsequent sections. In the training phase, for a particular IOS dental surface, direct sampling can be used to obtain the misaligned pointcloud, and GT pose transformation applied to the sampled result can yield the GT aligned pointcloud. Next, using these two pointclouds, we can extract their latent codes by Encoder, as shown by two burgundy data streams in Fig. 1. Then, using Projector, we determine the latent code of predicted aligned teeth based on the misaligned teeth, and we include a loss function to ensure that the latent codes of predicted aligned teeth are as compatible as possible with the GT aligned ones. Finally, we utilize Solver to infer two pose transformations based on corresponding differences, which derive from latent codes between misaligned teeth and predicted or GT aligned teeth, respectively. Furthermore, there are two loss functions in place to constrain these two pose estimates and GT pose transformation. Therefore, the total loss consists of two parts, one coming from the latent code predicted by the Projector and the other from two pose estimations of the Solver. However, in the inference phase, the GT aligned teeth are no longer required, as indicated by the data stream of wathet.

B. Preliminaries

Our method takes 3D IOSs as input, which is uniformly sampled as pointclouds using the Farthest Point Sampling [45] strategy. For each, the misaligned teeth pointcloud is denoted by $\mathcal{P} = \{\mathbf{P}_l \in \mathbb{R}^{n_p \times 3} | l \in \mathcal{L}\}$, where n_p denotes the number of sampled points on each tooth. The set $\mathcal{L}$ denotes the tooth labels according to FDI two-digit notation,[1] i.e. $\mathcal{L} = \{11-17, 21-27, 31-37, 41-47\}$ represents the FDI notation for 28 permanent teeth. Let $n_t = |\mathcal{L}|$ denotes the number of teeth in each case. Note that n_t is decided to be 28 in our dataset because most patients lack four third-molars, i.e. FDI number $\{18, 28, 38, 48\}$. The corresponding pointcloud of GT aligned teeth, which are designed by professional orthodontists, is denoted by $\tilde{\mathcal{P}} = \{\tilde{\mathbf{P}}_l \in \mathbb{R}^{n_p \times 3} | l \in \mathcal{L}\}$.

In our method, each tooth is regarded as a rigid body, and the alignment is defined as a 6-DOF (degree of freedom) pose transformation. For each tooth, we use two matrices to represent its rotation and translation. In particular, we use $\mathcal{Q} = \{\mathbf{q}_l \in$

[1] https://en.wikipedia.org/wiki/FDI_World_Dental_Federation

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

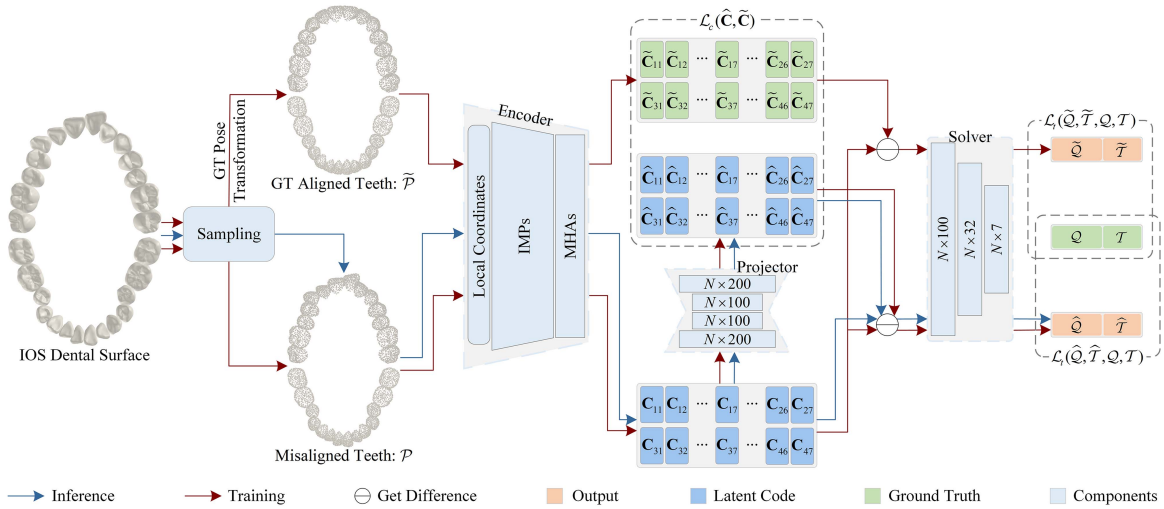


Fig. 1. The pipeline of LETA. In the inference phase, given a pointcloud of misaligned teeth $\mathcal{P}$ sampling from the IOS dental surface, we first use a feature Encoder to extract the latent code $\mathbf{C}$ from $\mathcal{P}$. Then, a Projector is applied to transform $\mathbf{C}$ into the latent code $\hat{\mathbf{C}}$, which is expected to be as close as possible to the latent code of GT aligned teeth. Finally, a Solver is used to infer the rotation $\hat{\mathcal{Q}}$ and translation $\hat{\mathcal{T}}$ based on the difference of $\mathbf{C}$ and $\hat{\mathbf{C}}$. In the training phase, we can get the GT aligned teeth $\tilde{\mathcal{P}}$ by applying the pose transformation of GT. Next, we use the same Encoder to extract the latent code $\tilde{\mathbf{C}}$ from $\tilde{\mathcal{P}}$, and use the same Solver to infer $\tilde{\mathcal{Q}}$ and $\tilde{\mathcal{T}}$ from $\mathbf{C}$ and $\tilde{\mathbf{C}}$. "Dashed boxes" indicate the loss we used. IMPs denote four-layer improved PointConv. MHAs denote three-layer multi-head attention.

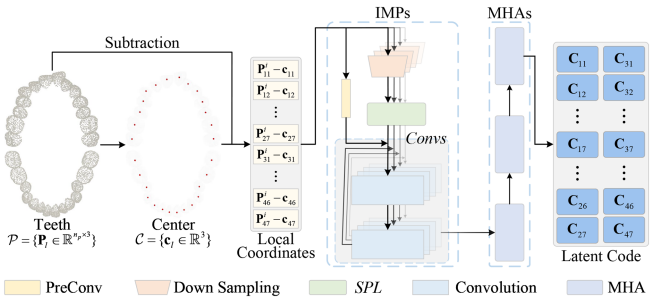


Fig. 2. The architecture of the Encoder in LETA. IMPs are used to extract local shape features, and MHAs are used to extract global context features. *SPL*: Shared Position Layer. MHA: Multi-head Attention. IMP: Improved PointConv.

$\mathbb{R}^4 | l \in \mathcal{L}\}$ to represent the rotation of each tooth, where $\mathbf{q}_l$ is a unit quaternion. We set the centroid of $\mathbf{P}_l$ as its rotation center, which is defined as:

$$\mathbf{c}_l = \frac{1}{n_p} \sum_{i=1}^{n_p} \mathbf{P}_l^i, \tag{1}$$

where n_p denotes the number of sampled points of $\mathbf{P}_l$ and $\mathbf{P}_l^i$ denotes the i-th point of $\mathbf{P}_l$. Similarly, we use $\mathcal{T} = \{\mathbf{t}_l \in \mathbb{R}^3 | l \in \mathcal{L}\}$ to represent the translation of each tooth.

C. Encoder

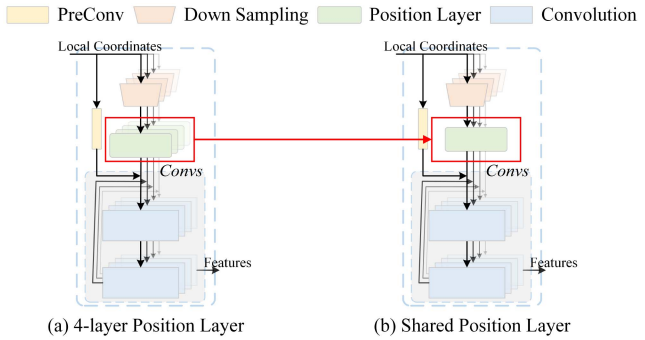
In the tooth alignment task, both local shape features and global context features are important because we should consider performance on both local occlusion and overall dental arch curve. Therefore, we build local and global feature extracting blocks in our Encoder, with pipeline illustrated in Fig. 2. Before these blocks, we obtain the center of each tooth based on (1) and calculate their local coordinates by subtraction with the corresponding center. Then, the first block takes the local coordinates of sampled points as input, and uses multi improved PointConv [20] layers called IMPs where we combine some redundant structures in different layers to extract local shape features, which ensures that LETA can learn detailed information from each individual tooth. We also stack several downsampling layers in the Encoder to gradually gather hierarchical multi-level local features. The second block uses multi-head attention layers [21] called MHAs to gather local features and extract global context features, which enables LETA to learn information such as teeth adjacency, occlusion and holistic dentition more effectively.

1) Local Shape Feature Extraction: The network we employed to extract local shape features is inspired by PointConv layers [20]. Those designed blocks will first gather local geometric information for each sampled point from its k nearest neighbors, then map this information to high-dimensional latent spaces, and finally update the corresponding local feature with convolutional layers.

PointConv makes use of non-uniform sampling to approximate a continuous convolution operation. For each sampled point, PointConv aggregates its local shape information from the difference (i.e. local coordinates) to its k nearest neighbors. However, recognizing that the value of local coordinates is only related to the relative positions of the current points pair and remains invariant to distinct layers, we propose to merge these aggregation processes across various layers, as shown in Fig. 3. Based on this idea, we use 4 consecutive IMP layers with decreasing sampling rates in this part. And multi-scale local shape features are accumulated in this way.

Specifically, in the t-th IMP layer, we first obtain k nearest neighbor points of every sampled point for teeth $\mathbf{P}_l$. We define the set of neighbours of the i-th point $\mathbf{P}_l^i$ in pointcloud $\mathbf{P}_l$

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.



(a) 4-layer Position Layer (b) Shared Position Layer

Fig. 3. Improvement in IMPs. Position Layer is used to aggregate features based on local coordinates.

as $\mathcal{N}(\mathbf{P}_l^i)$. Then, we use a multi-layer perceptron (MLP) to approximate the aggregation function (i.e. Position Layer) of neighbors with respect to the referenced point. Lastly, we employ a two-layer pointwise convolution operator ($Convs$) transforms features, i.e. aggregated features $MLP(\{\mathbf{P}_l^j - \mathbf{P}_l^i | \mathbf{P}_l^j \in \mathcal{N}(\mathbf{P}_l^i)\})$ and a $d^{(t-1)}$-dimensional vector, to the output feature, i.e. a $d^{(t)}$-dimensional vector. The computation process is defined as follows:

$$\mathbf{F}_l^{(t)} = \sigma\left(\sum_{\mathbf{P}_l^j \in \mathcal{N}(\mathbf{P}_l^i)} Convs^{(t)}(MLP^{(t)}(\mathbf{P}_l^j - \mathbf{P}_l^i), \mathbf{F}_l^{(t-1)})\right), \tag{2}$$

where $\mathbf{F}_l^{(t-1)} \in \mathbb{R}^{d^{(t-1)}}$ and $\mathbf{F}_l^{(t)} \in \mathbb{R}^{d^t}$ are the input and output features of the t-th IMP layer. Note that there is no $\mathbf{F}_l^{(t-1)}$ in the first IMP layer, so we use a convolution layer (PreConv) to initialize the feature as $\mathbf{F}_l^0$. $MLP^{(t)}$ maps $\mathbf{P}_l^j - \mathbf{P}_l^i$ into a d_{mid}-dimensional vectors, and we set $d_{mid} = 32$ for all layers. $Convs^{(t)} : (\mathbb{R}^{d_{mid}}, \mathbb{R}^{d^{(t-1)}}) \mapsto \mathbb{R}^{d^{(t)}}$ maps feature aggregated from relative position and output feature from the previous layer into a convolution kernel, and σ denotes the activation and batch normalization layers. Note that in this equation, $MLP^{(t)}$ of different layers has the same input $\mathbf{P}_l^j - \mathbf{P}_l^i$, which is independent of t. Thus we can merge these $MLP^{(t)}$ in various layers, resulting in:

$$\mathbf{F}_l^{(t)} = \sigma\left(\sum_{\mathbf{P}_l^j \in \mathcal{N}(\mathbf{P}_l^i)} Convs^{(t)}(SPL(\mathbf{P}_l^j - \mathbf{P}_l^i), \mathbf{F}_l^{(t-1)})\right), \tag{3}$$

where SPL, denoting Shared Position Layer, is a multi-layer pointwise convolution shared by different IMP layers which has the same effect with $MLP^{(t)}$. Therefore, $Convs^{(t)}$ in every layer aggregates the features from this layer $SPL(\mathbf{P}_l^j - \mathbf{P}_l^i)$ and previous layer $\mathbf{F}_l^{(t-1)}$. Finally, the output feature of $Convs^{(4)}$ will be transformed to MHAs for global context feature gathering.

2) Global Context Feature Extraction: Besides local shape features, the global relationship among different teeth is also of high necessity to determine the tooth alignment treatment plan. One way to do this is to manually define a hand-crafted relationship graph based on the orthodontic experience [11] and then use a graph-based method to extract context features. However, such a process will be very time-consuming and not all orthodontic constraints can be explicitly specified or defined. Thus, we use another method by employing the attention mechanism [21] to automatically learn the implicit contextual relationship between teeth. In our network, we take the output of the last IMP layer $\mathbf{F}_l^4$ as input, and feed it to MHAs to generate latent code $\mathbf{C}$:

$$\mathbf{C} = \mathbf{MHA}\left(\bigcup_{l\in\mathcal{L}} \mathbf{F}_l^4\right) = \bigcup_{i\in h} \mathbf{Att}_i\left(\bigcup_{l\in\mathcal{L}} \mathbf{F}_l^4\right) \tag{4}$$

$$= \bigcup_{i\in h} \sigma(\mathbf{Q}_i \mathbf{K}_i^T / \sqrt{d_k}) \mathbf{V}_i, \tag{5}$$

where $\mathbf{MHA}$ represents a multi-head attention layer. $\bigcup$ represents the union operation. The $\mathbf{Att}_i$ $(i = 1, 2, \ldots h)$ is the output of the i-th attention head while h is the number of heads. $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ represent the queries, keys and values which are obtained by applying a MLP layer on $\bigcup_{l\in\mathcal{L}} \mathbf{F}_l^4$[21]. d_k is the length of $\mathbf{Q}$. σ is a softmax layer. Finally, the result $\mathbf{C}$ of the last $\mathbf{MHA}$ is the latent code output by the Encoder.

D. Projector

It is noticed that compared with the misaligned teeth which vary a lot among patients, the aligned teeth are relatively stereotypical as they need to follow the orthodontic treatment rules, such as the "Andrew's six keys" [18]. With this intuition, after the latent code of misaligned teeth is extracted, we hope the Projector can regress it toward the predicted aligned one. In particular, the Projector takes the latent code of misaligned teeth $\mathbf{C}$ as input, and outputs the latent code of predicted aligned teeth $\hat{\mathbf{C}}$, which is encouraged to be close to the latent code of GT aligned teeth $\tilde{\mathbf{C}}$ during training. The computation is defined as below:

$$\hat{\mathbf{C}} = P(\mathbf{C}), \tag{6}$$

where P denotes the Projector network, which is composed of multiple fully connected layers. By doing so, our model can take advantage of the features from the GT aligned teeth, which guarantees its prediction with greater accuracy because of GT aligned teeth's better aesthetical performance and robust clinical applicability.

E. Solver

In LETA, the Solver is designed to regress the rigid body transformation matrices between the initial teeth and the target teeth with their latent codes' difference. In the training phase, the Solver takes the difference between latent codes of GT and misaligned teeth, i.e. $\tilde{\mathbf{C}}$ and $\mathbf{C}$, and the difference between latent codes of predicted teeth $\hat{\mathbf{C}}$ and misaligned teeth $\mathbf{C}$ as input, then regresses the rotation $\tilde{\mathcal{Q}}, \hat{\mathcal{Q}}$ and translation $\tilde{\mathcal{T}}, \hat{\mathcal{T}}$ respectively. In the inference phase, the latent code of GT aligned teeth becomes unavailable, so the Solver only takes the latent codes of misaligned teeth and predicted aligned teeth to regress.

As for the architecture details, we will first compute the difference between two latent codes. Then, it uses multi-layer pointwise convolutions to predict the rotation and translation of

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

each tooth. The computation is defined as:

$$(\hat{\mathcal{Q}}, \hat{\mathcal{T}}), (\tilde{\mathcal{Q}}, \tilde{\mathcal{T}}) = S(\ominus(\hat{\mathbf{C}}, \mathbf{C})), S(\ominus(\tilde{\mathbf{C}}, \mathbf{C})), \tag{7}$$

where S represents our Solver, $\ominus$ denote difference operation, $\tilde{\mathcal{Q}}$ and $\tilde{\mathcal{T}}$ represent the predictions of our Solver from $\tilde{\mathbf{C}}$ and $\mathbf{C}$ which are only used during training, while $\hat{\mathcal{Q}}$ and $\hat{\mathcal{T}}$ represent the predictions of our Solver from $\hat{\mathbf{C}}$ and $\mathbf{C}$ which are used for both training and inference.

F. Loss Function

As shown in the "dashed boxes" in Fig. 1, the loss of LETA contains two parts: latent code loss and transformation losses. The latent code loss measures the difference between the Projector's output $\hat{\mathbf{C}}$ and the latent code $\tilde{\mathbf{C}}$ of GT aligned teeth. As mentioned above, our model is supposed to learn effective geometric features from GT aligned teeth to guide learning, which can be realized by pushing two latent codes to be as close as possible. The transformation losses measure the performance of our Solver for predicting the rotation and translation matrices from the latent codes.

1) Latent Code Loss: When transforming $\mathbf{C}$ to $\hat{\mathbf{C}}$, the Projector expects the difference between $\hat{\mathbf{C}}$ and $\tilde{\mathbf{C}}$ to be as small as possible:

$$\mathcal{L}_c(\hat{\mathbf{C}}, \tilde{\mathbf{C}}) = \sum_{l \in \mathcal{L}} \sum_{i=1}^{n_p} \frac{\|\hat{\mathbf{C}}_l^i - \tilde{\mathbf{C}}_l^i\|^2}{\|\hat{\mathbf{C}}_l^i\| \|\tilde{\mathbf{C}}_l^i\|}, \tag{8}$$

where the purpose of $\|\hat{\mathbf{C}}_l^i\|$ and $\|\tilde{\mathbf{C}}_l^i\|$ in the denominator is to prevent this loss from being simply decreased due to the reduced norm of latent codes.

2) Transformation Losses: There are two pairs of rotation and translation prediction in the training phase, as mentioned in (7): 1) $\hat{\mathcal{Q}} = \{\hat{\mathbf{q}}_l | l \in \mathcal{L}\}$ and $\hat{\mathcal{T}} = \{\hat{\mathbf{t}}_l | l \in \mathcal{L}\}$ are predicted from $\mathbf{C}$ and $\hat{\mathbf{C}}$; 2) $\tilde{\mathcal{Q}} = \{\tilde{\mathbf{q}}_l | l \in \mathcal{L}\}$ and $\tilde{\mathcal{T}} = \{\tilde{\mathbf{t}}_l | l \in \mathcal{L}\}$ are predicted from $\mathbf{C}$ and $\tilde{\mathbf{C}}$. Based on these two predictions and the ground truth, we define two transformation losses as below:

$$\mathcal{L}_t(\hat{\mathcal{Q}}, \hat{\mathcal{T}}, \mathcal{Q}, \mathcal{T}) = \sum_{l \in \mathcal{L}} \|\hat{\mathbf{t}}_l - \mathbf{t}_l\|^2 + \lambda_q \sum_{l \in \mathcal{L}} (1 - \langle \hat{\mathbf{q}}_l, \mathbf{q}_l \rangle^2), \tag{9}$$

$$\mathcal{L}_t(\tilde{\mathcal{Q}}, \tilde{\mathcal{T}}, \mathcal{Q}, \mathcal{T}) = \sum_{l \in \mathcal{L}} \|\tilde{\mathbf{t}}_l - \mathbf{t}_l\|^2 + \lambda_q \sum_{l \in \mathcal{L}} (1 - \langle \tilde{\mathbf{q}}_l, \mathbf{q}_l \rangle^2), \tag{10}$$

where $\mathcal{Q}$ and $\mathcal{T}$ are rotation and translation of ground truth, $\langle \cdot, \cdot \rangle$ represents inner product operation, and the constant 1 guarantees the loss to be non-negative. For two unit quaternions, their inner product is the cosine of their vectorial angle. λ_q is a hyperparameter controlling the weight of quaternion loss.

3) Overall Loss: Our overall loss function is defined as the weighted sum of latent code loss and two transformation losses, which is mathematically defined by below:

$$\mathcal{L} = \lambda_c \mathcal{L}_c(\hat{\mathbf{C}}, \tilde{\mathbf{C}}) + \hat{\lambda}_t \mathcal{L}_t(\hat{\mathcal{Q}}, \hat{\mathcal{T}}, \mathcal{Q}, \mathcal{T}) + \tilde{\lambda}_t \mathcal{L}_t(\tilde{\mathcal{Q}}, \tilde{\mathcal{T}}, \mathcal{Q}, \mathcal{T}), \tag{11}$$

where $\lambda_c, \hat{\lambda}_t, \tilde{\lambda}_t$ are hyperparameters controlling the weight of each part, which are determined by a hyperparameter search procedure.

G. Implementation and Training Details

1) Network Details: In the Encoder, the dimensions of input and output are 3 and 200, respectively. The output of PreConv is 64. $n_p = 400$ points are sampled uniformly from each IOS dental surface. The number of points fed into 4-layer IMPs are 400, 100, 50 and 25 respectively. The output dimensions of all IMPs F_l are 200. The dimensions of output features of SPL d_{mid} in (3) are all set to be 32. All $Convs$ consist of two 1×1 convolution layers with output dimensions of 200. As for MHAs, the number of heads is equal to 5, and both the input dimension and output dimension are 200. We use the ReLU as activation function for the network except the output layer of Solver, whose activation is the identity function. The Projector consists of 4 fully connected layers, of which the output dimension are 200, 100, 100 and 200. The Solver is composed of 3 consecutive fully connected layers with output dimensions 100, 32 and 7 (4 for quaternion and 3 for translation estimation). The quaternion output of the Solver is divided by its norm to ensure it is a unit quaternion. We use batch renormalization [46] for the IMPs and MHAs in the Encoder because of small batch sizes during training. We also put the normalization layers forward [47] to accelerate training.

2) Training Details: We use the Adam optimizer to minimize the loss function. The whole network is trained in an end-to-end manner. We set $\lambda_q = 100$, $\lambda_c = 3$, $\hat{\lambda}_t = 0.3$, and $\tilde{\lambda}_t = 1$ in the loss function. The network is trained from scratch with a batch size of 4 and a learning rate of 0.0001 using 100 epochs. To augment the training data, we first randomly replace 30% of the misaligned teeth with their corresponding GT aligned counterparts. Next, we rotate each tooth by a random angle, i.e. $[0°, 30°]$, in a random direction with a probability of 10% individually, and then translate it by a random distance, i.e. $[0\text{mm}, 3\text{mm}]$, in a random direction with a probability of 10% individually. Since the output of our network is the rotation and translation, we must modify the ground truth by the corresponding amount when transforming the input teeth.

Our network is implemented with Tensorflow and trained on a server with an NVIDIA GeForce RTX 3090 GPU and Ubuntu 20.04 OS. In the training phase, it takes about 34 hours for the whole 100 epochs. In the inference phase, it takes about 0.013 seconds on average for transformation prediction given the upper and lower jaw misaligned teeth from one patient.

IV. Experiments

In this section, we first present the dataset and evaluation metrics. Next, we compare the proposed LETA with previous state-of-the-art tooth alignment methods. Then, we present detailed ablation experiments and a clinical applicability study to demonstrate the effectiveness of our method. Finally, we have a further investigation on our LETA about its strengths and limitations.

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

A. Dataset

Our dataset consists of 9,868 pairs of misaligned 3D IOS teeth surfaces and the corresponding rigid transformations of each tooth for orthodontics treatment planning. 28 teeth are contained in each pair of data. All treatment plans are designed by professional orthodontists with years of experience. All the samples are real-world clinical treatment cases collected from dental hospitals in China. We randomly divide the 9,868 pairs into a training set of 8,913 pairs and a testing set of 955 pairs, respectively.

B. Evaluating Metrics

To evaluate the performance of our model, we use the following three metrics: rotation error, translation error and point-wise distance. Rotation error and translation error illustrate the performance of 6-DOF rigid body transformation for each tooth. Point-wise distance evaluates holistic differences for corresponding points between two dentition pointclouds.

Rotation error: It measures the average angular error for one patient based on every tooth pair, which is defined as:

$$\theta(\hat{\mathcal{Q}}, \mathcal{Q}) = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \theta(\hat{\mathbf{q}}_l, \mathbf{q}_l), \tag{12}$$

where $\hat{\mathcal{Q}}$ and $\mathcal{Q}$ are predicted and ground truth rotation, $\hat{\mathbf{q}}_l$ and $\mathbf{q}_l$ are for one single tooth. $\theta(\hat{\mathbf{q}}_l, \mathbf{q}_l)$ represents angular error and can be calculated with the following formula:

$$\theta(\hat{\mathbf{q}}_l, \mathbf{q}_l) = 2 \arccos(|\langle \hat{\mathbf{q}}_l, \mathbf{q}_l \rangle|), \tag{13}$$

where $\langle \cdot, \cdot \rangle$ means inner product.

Translation error: It measures the displacement error by taking the average mean square error on the translation of every tooth pair, which is defined as:

$$\delta(\hat{\mathcal{T}}, \mathcal{T}) = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \|\hat{\mathbf{t}}_l - \mathbf{t}_l\|^2, \tag{14}$$

where $\hat{\mathcal{T}}$ is predicted translation and $\mathcal{T}$ is ground truth translation. Similarly, $\hat{\mathbf{t}}_l$ and $\mathbf{t}_l$ are specific to one single tooth.

Point-wise distance: It measures the average L2 distance between predicted and GT aligned teeth on every point-pair, which is defined as:

$$d(\hat{\mathcal{P}}, \tilde{\mathcal{P}}) = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \left(\frac{1}{|n_p|} \sum_{i=1}^{n_p} \|\hat{\mathbf{P}}_l^i - \tilde{\mathbf{P}}_l^i\|^2 \right), \tag{15}$$

where $\hat{\mathcal{P}}$ and $\tilde{\mathcal{P}}$ are pointclouds of predicted and GT aligned teeth, respectively. $\hat{\mathbf{P}}_l^i$ and $\tilde{\mathbf{P}}_l^i$ denote the corresponding i-th point in each tooth pair.

C. Comparisons

To evaluate the effectiveness of our LETA, we conduct comparisons with advanced pointcloud-based frameworks PointNet++ [29], PointConv [20], RSCNN [48] and DGCNN [35], and state-of-the-art automatic tooth alignment PSTN [10], TANet [11] and TAligNet [12]. For a fairer comparison with them, we slightly modify their output layers in the form of our method to conduct the comparison. For PointNet++, we adapt the classification-focused design and set the output dimension of the final linear layer to 7 for representing the pose transformation. Similarly, we use the classification-based PointConv, RSCNN and DGCNN by adjusting their last linear layers' output to 7. For PSTN, we add a fully connected layer following the Localization Network to get predicted transformation with dimension 7. As for TANet, we remove the module Tooth Assembler, which is used to transform, assemble and generate the final aligned pointcloud, then output the predicted pose directly. For TAligNet, we define the rotation center of each tooth as its own centroid and the number of sampling points of each tooth $n_p = 400$, which are also applied to other methods.

As reported in Table I, LETA can achieve the mean rotation error of 5.28°, translation error of 0.91mm, and point-wise distance of 0.97mm, which are significantly better than all the baseline methods, e.g. 0.44°, 0.10mm and 0.10mm improvements to TAligNet. At the same time, LETA performs better in all three rotation error dimensions, particularly in the z-dimension, where has a noticeable improvement of 0.28° to sub-optimal TAligNet. We further display the standard deviation of prediction errors about eight methods in Table I. As we can see, LETA outperforms all other seven methods, e.g. the overall rotation error deviation of LETA is 2.95°, which is 0.31° lower than TAligNet, the translation error deviation is 0.56mm, which is 0.03mm lower than the 0.59mm of TAligNet, the point-wise distance deviation is 0.57mm, which is 0.04mm lower than the 0.61mm of TAligNet, which means the prediction of our methods is more robust. Therefore, the smallest mean error and the lowest standard deviation both indicate that the results of our method are more evenly distributed, and performance is more consistent.

We further illustrate the superiority of our method with qualitative visualizations. Three visualization cases with crowded dentition are presented in Fig. 4 including our LETA and other compared methods. We employ the point-wise distance heat map to emphasize the discrepancy between the predicted results and the GT. As illustrated, the results of other methods have lots of areas with large point-wise distance, while the result of LETA infrequently commits such errors. In addition, the results of the tooth alignment obtained by compared methods are visually in striking contrast to those of LETA in the challenging areas highlighted by the "red boxes". To be specific, the lateral incisor and cuspid in the first row of the upper jaw exhibit a relatively low point-wise distance with respect to GT by LETA, while other methods still have a severe crowd (e.g. TANet) or seldom improve the alignment (e.g. PointNet++). The lateral incisor in the second row of the maxilla can be effectively aligned using LETA, while TAligNet retains a minor error, TANet leaves a gap between the lateral incisor and central incisor, and other methods even cannot move it above the dental arch. As for "ellipses" in the third row of Fig. 4, it is abundantly clear that when processing teeth with malocclusion, LETA guarantees proper positions for the rest of teeth in dentition, whereas other methods introduce a large point-wise distance to them. The reason for above mentioned is that we can fully extract the features of both misaligned and GT aligned teeth during training, leading to great

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISONS BASED ON THE MEAN AND STANDARD DEVIATION OF ROTATION ERROR, TRANSLATION ERROR AND POINT-WISE DISTANCE

Model		Rotation Error (degree)				Translation Error (mm)	Point-wise Distance (mm)
		R_{xyz}	R_x	R_y	R_z		
PointNet++	Mean	11.79	4.47	5.08	7.71	1.87	1.99
	Std Dev	8.80	4.58	5.23	7.99	1.08	1.15
PointConv	Mean	11.80	4.48	5.07	7.72	1.87	1.99
	Std Dev	8.80	4.58	5.24	7.98	1.07	1.13
RSCNN	Mean	9.11	3.35	3.75	6.21	1.58	1.68
	Std Dev	6.71	3.23	3.91	6.35	0.98	1.02
DGCNN	Mean	6.14	2.78	2.69	3.64	1.31	1.37
	Std Dev	3.70	2.42	2.46	3.46	0.80	0.81
PSTN	Mean	8.19	3.48	3.76	4.93	1.58	1.65
	Std Dev	4.94	3.02	3.36	4.63	0.94	0.97
TANet	Mean	6.62	3.09	3.12	3.69	1.14	1.20
	Std Dev	3.91	2.77	2.79	3.42	0.65	0.67
TAligNet	Mean	5.72	2.80	2.67	3.07	1.01	1.07
	Std Dev	3.26	2.35	2.33	2.91	0.59	0.61
LETA	Mean	**5.28**	**2.59**	**2.51**	**2.79**	**0.91**	**0.97**
	Std Dev	**2.95**	**2.20**	**2.19**	**2.57**	**0.56**	**0.57**

R_{xyz} denotes the overall rotation error. R_x, R_y and R_z denote the rotation error in the dimension of x, y and z, respectively.

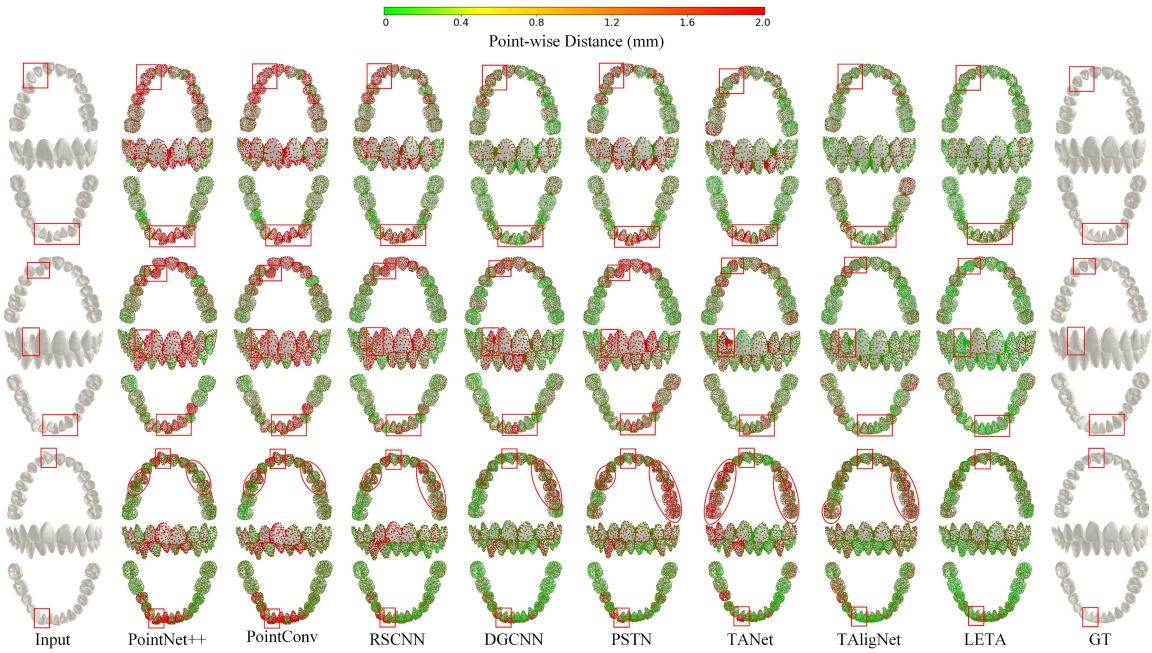


Fig. 4. Qualitative comparison between PointNet++, PointConv, RSCNN, DGCNN, PSTN, TANet, TAligNet and LETA. The point-wise distance between their predictions and GT is used to enhance the visualization. "Boxes" denote teeth with severe malocclusion. "Ellipses" denote large point-wise distance other teeth in the dentition have due to orthodontia.

performance in latent code projection and pose transformation estimation.

D. Ablation Study

1) Architecture-Based Ablation: In order to test the effectiveness of the Encoder, Projector and the architecture of dual-branch latent encoding, as well as IMPs and MHAs, we designed nine different versions using various parts of the LETA. Their performance is shown in Table II, and the complete version of LETA is displayed in the last row.

Ablation on the structure of model: The results of this ablation study are displayed in initial three rows of Table II:

- The first row denotes a version without the Encoder and Projector. We employed a simple MLP replacing the Encoder to aggregate the features of the $\mathcal{P}$ and $\tilde{\mathcal{P}}$. The

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

TABLE II
ABLATION STUDY ON WHETHER OR NOT TO USE THE ENCODER, PROJECTOR, OR ENCODE THE GT ALIGNED TEETH, AS WELL AS THE USE OF IMPS OR MHAS IN THE ENCODER

GT Aligned	Encoder IMPs	Encoder MHAs	Projector	Solver	Rotation Error (degree)	Translation Error (mm)	Point-wise Distance (mm)	Model Size (MB)
✓				✓	8.22(↑0.80)*	1.32(↑0.15)*	1.43(↑0.17)*	1.82*
✓			✓	✓	6.81(↑0.29)†	1.13(↑0.03)†	1.21(↑0.03)†	3.12†
✓	✓	✓		✓	5.70	0.96	1.02	17.77
				✓	7.42*	1.17*	1.26*	1.82*
			✓	✓	6.52†	1.10†	1.18†	3.12†
	✓	✓	✓	✓	5.32‡	0.93‡	0.99‡	19.07‡
✓		✓	✓	✓	11.63	1.54	1.68	11.64
✓	✓		✓	✓	5.68	1.08	1.14	10.54
✓	PointConv	✓	✓	✓	**5.27**	0.92	**0.97**	19.16
✓	✓	✓	✓	✓	5.28(↓0.04)‡	**0.91**(↓0.02)‡	**0.97**(↓0.02)‡	19.07‡

Special symbols (* †‡) are used to distinguish models of varying size. Values in parentheses indicate the performance impact of encoding GT aligned teeth under the same model size. "PointConv" denotes the substitution of IMPs with PointConv.

outcomes of both are then directly fed into the Solver to predict pose transformations. Consequently, the loss function used to train the entire network encompasses only transformation losses in (9) and (10).

- In the second row, we replaced the Encoder with a simple MLP to map the dimensions of pointclouds to the Projector's input space. Apart from this change, all other aspects, including the loss function, remain identical to the complete version of LETA.
- In the third row, for the version without the Projector, we directly engaged the Solver to predict pose transformations based on $\mathbf{C}$ and $\tilde{\mathbf{C}}$, respectively. Therefore, the input for the Solver is no longer the difference between various latent codes, but the latent codes themselves. Additionally, the loss function will only comprise the two transformation losses.

Comparing these three versions with the full version of LETA, both the Encoder and Projector contribute to significant improvements. Notably, when both are removed, there is a deterioration of 2.94° in rotation error, 0.41mm in translation error, and 0.45mm in point-wise distance. Therefore, the Encoder which integrates IMPs extracting local features and MHAs capturing global features will aid in harnessing more information from the dental pointcloud to achieve better results. Meanwhile, it is reasonable to use the Projector to generate the latent code for predicted aligned teeth and to guide pose transformations based on the differences between latent codes.

Ablation on the structure of dual-branch framework: The results of this ablation study are displayed in 4-6 rows of Table II. They all do not introduce GT aligned teeth, meaning that the data flow during both the training and inference stages is completely identical, and the loss function only includes $\mathcal{L}_t(\hat{\mathcal{Q}}, \hat{\mathcal{T}}, \mathcal{Q}, \mathcal{T})$.

- The fourth row is a version that excludes GT aligned teeth from the first row. We use a simple MLP to aggregate features from $\mathcal{P}$ and then fed them into the Solver for pose prediction.
- The fifth row is similar to the second row. We use a simple MLP to replace the Encoder for feature mapping, but without the branch for processing GT aligned teeth and excluding associated loss functions.
- Different from the complete version, in the sixth row, we use the Encoder to learn latent code $\mathbf{C}$ from $\mathcal{P}$, and predict $\hat{\mathbf{C}}$ based on the Projector. Then, we employ the Solver to infer $\hat{\mathcal{Q}}$ and $\hat{\mathcal{T}}$ from the difference between $\hat{\mathbf{C}}$ and $\mathbf{C}$.

As shown in Table II, with the increase in model size, the overall performance of tooth alignment gradually improves. Meanwhile, introducing geometric information and clinical rules from GT aligned teeth has a negative impact when the model size is small. However, this impact gradually diminishes as parameters increases, disappearing completely and even resulting in a slight improvement when the scale reaches 19.07MB. In fact, the information learnt from GT aligned teeth often manifest in extremely subtle forms. Therefore, a powerful and effective feature extraction and processing module is the core of our dual-branch latent encoding framework. This also inspires the direction for further optimization of our work in the future.

Ablation on the substructure of Encoder: The results of this ablation study are displayed in 7-9 rows of Table II:

- In the seventh row, we used simple fully connected layers to replace IMPs for mapping the pointcloud to the MHAs' input space.
- In the eighth row, we do not use MHAs to extract global features and directly treated the results obtained from IMPs as latent code.
- In the ninth row, we replace the IMP with PointConv for extracting local shape features.

As observed, the versions without IMPs or MHAs both perform quite poorer than the whole LETA, and IMPs contribute more significant improvement relative to MHAs. This is primarily due to the fact that the pose transformations obtained from LETA are tailored to specific teeth. As such, effectively learning the geometric information of individual teeth can contribute to improved results. However, neglecting individual teeth and only considering the entire dentition and context features would cause the pose transformations to be misaligned with their corresponding teeth, leading to an adverse effect on the results.

For the ninth row, compared with the complete version, PointConv increases the model size from 19.07MB to 19.16MB, but introduces only minor fluctuations in rotation error and translation error, as well as obtains a nearly identical overall

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

TABLE III
COMPARISON EXPERIMENT ON MODELNET40

Method	Accuracy (%)	Model Size (MB)
w/o *SPL*	85.7	7.7
with *SPL*	**86.0**	**7.6**

IMPs w/o SPL is depicted in Fig. 3(a), while with SPL is presented in Fig. 3(b).

TABLE IV
ABLATION STUDY WITH DIFFERENT SAMPLING POINTS IN ENCODER'S LOCAL FEATURE EXTRACTION LAYERS IMPS

Sampling Points	Rotation Error (degree)	Translation Error (mm)	Point-wise Distance (mm)
200	5.304	0.926	0.983
400	**5.278**	0.911	0.968
600	5.323	**0.904**	**0.961**

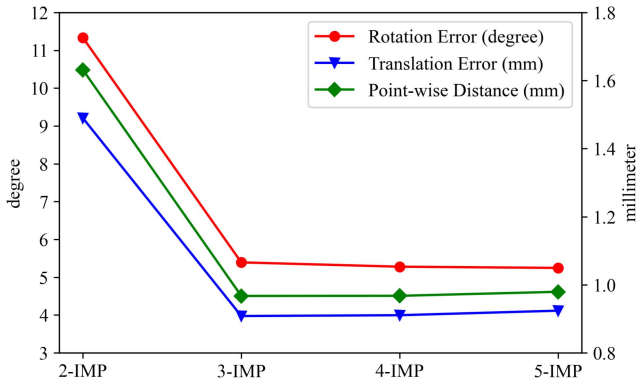


Fig. 5. Ablation Study with different number of IMPs' layer. "N-IMP" denotes N-layer IMPs.

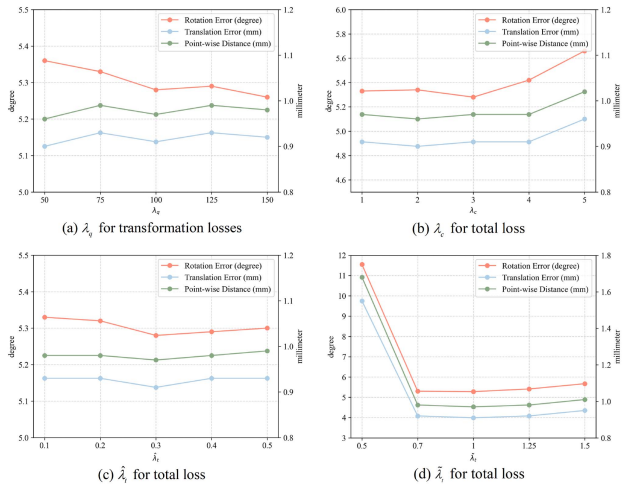


Fig. 6. Ablation study for parameters in the loss function.

performance. Furthermore, considering IMPs only encompasses encoding, we excluded experiments on pointcloud segmentation and conducted on classification task ModelNet40[49] to explore its generalization, as shown in Table III. We are able to reduce the model size from 7.7M to 7.6M by merging redundant structures, while achieving a slight improvement of 0.3%. Therefore, the results on ModelNet40 demonstrate not only IMPs' capability to optimize spatial complexity but also its potential applicability to other generic tasks.

2) Parameter-Based Ablation: The first we cover is the impact of 4-layer IMPs with decreasing sampling points on the performance under various sampling point choices. Then, the impact of utilizing IMPs with various layers on the prediction results is explored. Finally, we displayed the effect with different weights for the overall loss in (9), (10) and (11).

As shown in Table IV, three variants are used: the 4-layer IMPs with 200 sampling points for the mini-variant, the proposed LETA for the mid-variant, and the 4-layer IMPs with 600 sampling points for the large-variant. While the large-variant produces better results in translation error of 0.904mm and point-wise distance of 0.961mm, the mid-variant produces the greatest result in rotation error of $5.278°$. Actually, the differences in their performance of translation error 0.007mm and point-wise distance 0.007mm are not considerable, but the improvement of the mid-variant on rotation error $0.045°$ is relatively more impressive. Additionally, the mid-variant speeds up the inference of all testing data by more than 40%, taking only about 12.30 seconds compared to the large-variant's 21.17 seconds. Therefore LETA is constructed by 4-layer IMPs with sampling points of 400.

The influences of employing various IMP layers on the results are shown in Fig. 5. The notation, being used in this experiment, is 'N-IMP', which means the architecture with N-layer IMPs. As Fig. 5 illustrated, 3-IMP, 4-IMP and 5-IMP greatly outperform 2-IMP in all evaluating metrics. For 3-IMP and 4-IMP, 4-IMP outperforms 3-IMP by $0.1637°$ in rotation error, which is a comparably large improvement, while the improvements in translation error and point-wise distance are trivial. For 5-IMP, the rotation error has a $0.0326°$ decrease from $5.2776°$ to $5.2450°$, and the translation error and point-wise distance increase by 0.0133mm and 0.0118mm, which are all close to 4-IMP. However, an additional layer of IMP results in much larger space complexity and 12h longer training time. Thus, we decide on 4-layer IMPs.

3) Loss Function-Based Ablation: Furthermore, we also conducted ablation studies on hyperparameters and components in loss functions.

Hyperparameters in loss function: Fig. 6 delineates the ablation analysis associated with various hyperparameters in the loss function. Concerning λ_q for transformation losses in Fig. 6(a), we gravitated toward $\lambda_q = 100$ to ensure a harmonized equilibrium amongst all three metrics, although the rotation error reaches its nadir at $5.26°$ which is only $0.02°$ better than us when $\lambda_q = 150$, the translation error and point-wise distance register their minima at 0.9mm and 0.96mm respectively which both manifest an improvement of only 0.01mm when $\lambda_q = 50$. Similarly, when determining the λ_c for total loss in Fig. 6(b), $\lambda_c = 3$ provides superior results from a holistic perspective, even though the translation error and point-wise distance attain their minima at $\lambda_c = 2$, which we consider as a local optimum due to the deterioration of $0.06°$ in rotation error it introduces. For

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

TABLE V
ABLATION STUDY ON COMPONENTS OF OVERALL LOSS

Eq. (8)	Eq. (9)	Eq. (10)	Rotation Error (degree)	Translation Error (mm)	Point-wise Distance (mm)
	✓	✓	5.339	0.916	0.974
✓	✓		5.280	0.922	0.979
	✓		5.317	0.931	0.988
✓	✓	✓	**5.278**	**0.911**	**0.968**

TABLE VI
CLINICAL APPLICABILITY STUDY ON PERFORMANCE AND EFFICIENCY OF LETA AND BASELINES

Methods	Adjacent Relationship	Occlusion	Tooth Rotation	Molar Relationship	Workload (%)
PSTN	1.30	1.40	1.46	0.62	78.52 (↓21.48)
TANet	1.40	1.42	1.76	1.10	73.56 (↓26.44)
TAligNet	1.94	2.28	1.98	1.46	62.92 (↓37.08)
LETA	**3.02**	**3.40**	**3.04**	**2.80**	**39.04 (↓60.96)**

"Workload" indicates the percentage of time used by introducing automatic tooth alignment methods compared to starting from scratch, with the savings shown in parentheses.

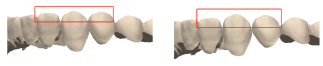
$\hat{\lambda}_t$ and $\tilde{\lambda}_t$ in Fig. 6(c) and (d), respectively, we directly selected $\hat{\lambda}_t = 0.3$ and $\tilde{\lambda}_t = 1$ where both points concurrently achieve optimal values across the three metrics.

Components in loss function: The experiment on various components of overall loss is illustrated in Table V. For the first row, we fixed the network architecture, only removing $\mathcal{L}_c(\hat{\mathbf{C}}, \tilde{\mathbf{C}})$ that guides the Projector's alignment between $\hat{\mathbf{C}}$ and $\tilde{\mathbf{C}}$ in latent space. For the second row, we removed the constraint on $\tilde{\mathcal{Q}}$ and $\tilde{\mathcal{T}}$ from (10). Thus, during the training phase, the network only encodes $\tilde{\mathcal{P}}$ to obtain $\tilde{\mathbf{C}}$, without predicting $\tilde{\mathcal{Q}}$ and $\tilde{\mathcal{T}}$ based on $\tilde{\mathbf{C}}$ and $\mathbf{C}$. For the third row, we retained only $\mathcal{L}_t(\hat{\mathcal{Q}}, \hat{\mathcal{T}}, \mathcal{Q}, \mathcal{T})$, whose setting is consistent with the fourth row in Table II. Comparing their performance with the full version of LETA in the last row, it is evident that both the Projector's alignment in latent space and the complete process of GT aligned teeth in dual-branch encoding are significant for our task.

E. Clinical Applicability Study

In order to evaluate the clinical usage of our method, we have conducted a clinical applicability study. We randomly selected 50 additional orthodontic cases (out of training and testing data, collected from real-world cases) and predicted the alignment plan with LETA and baselines [10], [11], [12]. We invited orthodontists with substantial experience to assess the performance of LETA and baselines from following perspectives and to further report the potentially reduced workload by different methods. The orthodontists were asked to score in a discrete scale from 1 to 5, with 5 indicating a perfect alignment, score 4 barely satisfied (qualified with only minor issues towards perfect), score 3 qualified, 2 barely qualified (there are lots of problems), and 1 unqualified. Finally, compared to starting from scratch, the orthodontists would assess the workload reduction assisted by these automatic tooth alignment methods.

As shown in Table VI, there are 4 aspects of performance scoring statistics:



(a) Bad (b) Good

Fig. 7. Examples of adjacent relationship. The central incisor should be slightly higher than adjacencies.



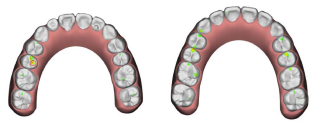
(a) Bad (b) Good

Fig. 8. Examples of occlusion. The upper incisors should slightly overlap the lower ones.



(a) Bad (b) Good

Fig. 9. Examples of tooth rotation. The lateral incisor and cuspid should be parallel to the adjacent teeth and neatly aligned along the dental arch.



(a) Bad (b) Good

Fig. 10. Examples of molar relationship. The red and green area indicate the deep and shallow collision with the opposite teeth, respectively.

- *Adjacent relationship:* A proper alignment not only considers the lateral proximity of teeth but also their vertical arrangement, which is intrinsically linked to their functional roles and positional hierarchy within the mouth. Fig. 7 demonstrates an example of good and bad adjacent relationships.
- *Occlusion:* It is essential to maintain normal vertical and horizontal contact depths, avoiding excessive overbite or overjet, as demonstrated in Fig. 8.
- *Tooth rotation:* In a healthy dentition, each tooth should be parallel to its neighbors and ensure alignment with the dental arch, as shown in Fig. 9.
- *Molar relationship:* As shown in Fig. 10(b), in a dentition with good molar relationship, the cusp of a molar should be aligned with the groove of the opposite molars. Fig. 10(a) shows an example of bad molar relationship, the cusp of a molar on the left has a deep collision with the cusp of the opposite molar. In addition, the collision depth of the left and right molars is inconsistent, and the right molar has almost no collision.

In this study, LETA consistently outperforms baselines across various aspects, surpassing qualified levels on three indicators, except for the molar relationship, which is reported as nearly

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

TABLE VII
FURTHER DISCUSSION ON DIFFERENT TEETH

Model	Tooth	Rotation Error (degree)				Translation Error (mm)	Point-wise Distance (mm)
		R_{xyz}	R_x	R_y	R_z		
PSTN	Central Incisor	7.65	4.17	2.23	4.84	1.99	2.02
	Lateral Incisor	9.17	4.71	3.48	5.48	1.94	1.99
	Cuspid	8.88	3.65	4.31	5.39	1.66	1.73
	First Bicuspid	8.55	3.09	4.33	5.29	1.40	1.49
	Second Bicuspid	8.65	3.03	4.04	5.68	1.44	1.53
	First Molar	6.69	2.56	3.55	3.84	1.26	1.35
	Second Molar	7.76	3.15	4.39	3.97	1.33	1.44
TANet	Central Incisor	6.11	3.43	2.49	3.22	1.27	1.31
	Lateral Incisor	7.22	3.75	3.24	3.89	1.28	1.32
	Cuspid	7.42	3.42	3.42	4.35	1.19	1.25
	First Bicuspid	6.79	2.98	3.12	4.01	1.05	1.12
	Second Bicuspid	6.92	2.97	3.28	4.04	1.09	1.16
	First Molar	5.15	2.22	2.54	2.88	1.00	1.07
	Second Molar	6.77	2.88	3.79	3.44	1.08	1.18
TAligNet	Central Incisor	4.40	2.88	1.61	2.02	1.07	1.09
	Lateral Incisor	5.63	3.15	2.50	2.82	1.08	1.11
	Cuspid	6.33	3.01	2.96	3.59	1.07	1.12
	First Bicuspid	6.22	2.86	2.82	3.60	0.95	1.01
	Second Bicuspid	6.34	2.88	2.91	3.66	0.96	1.04
	First Molar	4.74	2.05	2.37	2.64	0.92	0.99
	Second Molar	6.37	2.81	3.55	3.15	1.01	1.10
LETA	Central Incisor	**4.21 (↓0.19)**	2.76	1.51	1.97	**0.97 (↓0.10)**	**0.99 (↓0.10)**
	Lateral Incisor	**5.23 (↓0.40)**	3.03	2.32	2.54	**0.97 (↓0.11)**	**1.00 (↓0.11)**
	Cuspid	**5.76 (↓0.57)**	2.77	2.72	3.20	**0.94 (↓0.13)**	**0.99 (↓0.13)**
	First Bicuspid	**5.61 (↓0.61)**	2.54	2.64	3.19	**0.85 (↓0.10)**	**0.92 (↓0.09)**
	Second Bicuspid	**5.67 (↓0.67)**	2.55	2.71	2.30	**0.86 (↓0.10)**	**0.93 (↓0.11)**
	First Molar	**4.47 (↓0.27)**	1.95	2.29	2.44	**0.85 (↓0.07)**	**0.92 (↓0.07)**
	Second Molar	**6.00 (↓0.37)**	2.57	3.39	2.98	**0.93 (↓0.08)**	**1.02 (↓0.08)**

The improvements in "brackets" are compared with sub-optimal results, which are highligh ed by "underlines".

qualified. Remarkably, with the assistance of LETA, orthodontists only need 39.04% of the time to complete tooth alignment compared to starting from scratch, reducing workload over 60%. This highlights the potential practical implications of our method in greatly alleviating doctor-patient conflicts. Furthermore, although the quantitative improvements of LETA over the sub-optimal TAligNet in terms of rotation error, translation error, and point-wise distance are relatively limited as shown in Table I, its clinical performance is far superior. In fact, the proficiency of an orthodontist is frequently measured by their meticulous control over details, ensuring that deviations across various dimensions remain minimal. Therefore, when the performance of automatic tooth alignment methods reach a certain standard, even minor quantitative enhancements can translate into considerable clinical improvements.

F. Discussion

1) Analysis of Different Teeth: We present the performance of PSTN [10], TANet [11], TAligNet [12], and our LETA on seven different types of teeth (central incisor, lateral incisor, cuspid, first bicuspid, second bicuspid, first molar and second molar) in Table VII. Among the various teeth types, PSTN is the least effective. TAligNet delivers the second-best performance, while our LETA surpasses all other methods across all teeth categories.

Specifically, LETA shows the largest performance improvement on the cuspid and the least improvement on the first molar.

TABLE VIII
STATISTIC FOR EACH TYPE OF TEETH IN THE TESTING DATASET

Tooth	Rotation Error (degree)	Translation Error (mm)	Point-wise Distance (mm)
Central Incisor	14.15	2.55	2.71
Lateral Incisor	15.24	2.36	2.55
Cuspid	14.68	2.08	2.28
First Bicuspid	11.43	1.60	1.91
Second Bicuspid	11.34	1.62	1.98
First Molar	6.64	1.50	1.70
Second Molar	9.10	1.63	1.83

Referring to a statistic of rotation error, translation error and point-wise distance for each type of teeth in the testing dataset, the first molar has the lowest errors of $6.64°$, 1.50mm and 1.70mm, and the cuspid has higher errors of $14.68°$, 2.08mm and 2.28mm, as displayed in Table VIII. Therefore, we think there is a positive correlation between the severity of original malocclusion and the degree of performance improvement.

Moreover, LETA demonstrates superior performance on the first molar, which can be attributed to its relatively lower severity of malocclusion. In contrast, LETA performs less well on the lateral incisor, cuspid and second molar. Several reasons underlie these findings. The lateral incisor and cuspid, situated between the central incisor and molars, present complex anatomical and orthodontic challenges. The second molar, positioned at the terminus of the dentition, provides limited surrounding dental features for effective learning.

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

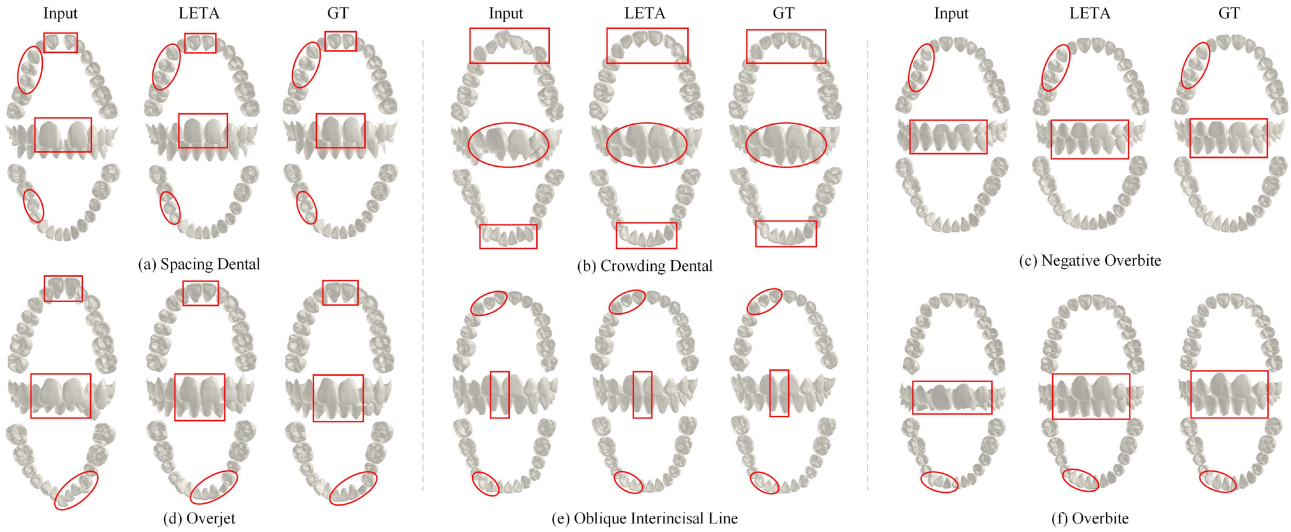


Fig. 11. More results about different malocclusion with LETA's good performance. "Boxes" denote the features that identify the type of malocclusion. "Ellipses" denote other misaligned teeth.

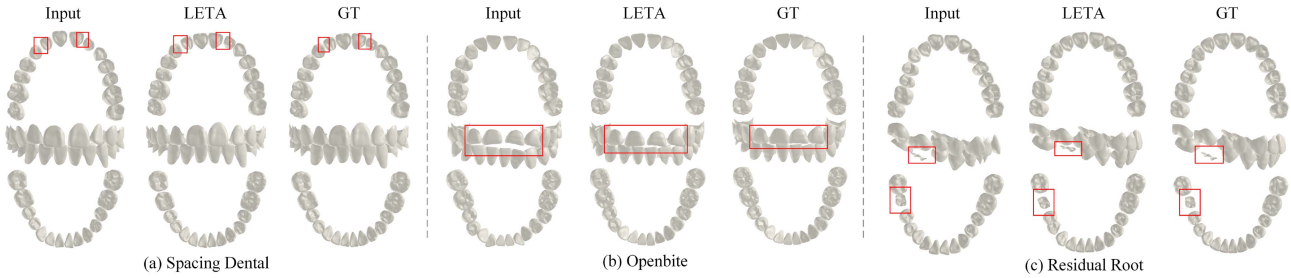


Fig. 12. More results about different malocclusion with LETA's bad performance. "Boxes" denote the features that identify the type of malocclusion. "Ellipses" denote gaps between teeth that were initially proximate. The dentition in the middle of (c) is a lateral view with occlusion.

2) Analysis of Different malocclusion: We apply LETA to various malocclusion, including the spacing dental, crowding dental, negative overbite, overjet, oblique interincisal line, and more. Then we illustrate that from Fig. 11 and Fig. 12, despite several limitations, LETA can generally perform well.

The following lists different types of malocclusion in Fig. 11 where LETA has demonstrated commendable performance.

- *Dental Spacing:* There is a pronounced gap between the two central incisors in Fig. 11(a), and LETA effectively aligns the dentition, substantially reducing the gap.
- *Dental Crowding:* The dentition from Fig. 11(b) exhibits extremely severe crowded alignment in both the maxilla and mandible, but LETA accurately predicts pose transformations for each tooth, yielding the result that closely aligns with the GT.
- *Negative Overbite:* Fig. 11(c) shows an instance of the negative overbite, in which the lower jaw wraps around the upper jaw in the occlusal condition. Finally, this issue is resolved and additional misalignment problems are fixed by LETA.
- *Overjet:* As opposed to overbite, overjet is characterized by protrusion of the upper central incisors, which causes an excessive horizontal distance between the upper and lower central incisors. The maxilla of the input in Fig. 11(d) clearly displays this trait. However, the result of LETA demonstrates that the issue is effectively resolved.
- *Oblique Interincisal Line:* With this type of malocclusion in Fig. 11(e), the midlines of upper central incisors and lower central incisors do not line up, when the teeth are occluded. Although it doesn't appear to have much of an impact, the significant oblique interincisal line can cause facial distortion. Therefore, we investigate LETA's performance on this malocclusion and obtain comparable results.
- *Overbite:* As illustrated in Fig. 11(f), overbite occurs when the upper jaw overlaps the lower jaw in occlusal condition. According to the final result, LETA substantially improves the occlusal relationship between the maxilla and mandible.

There are several malocclusion cases where our LETA cannot perform well, as illustrated below.

- *Dental Spacing:* In Fig. 12(a), which showcases a more severe dentition spacing, the result demonstrates that LETA can not deal with such difficult circumstances. Specifically, within the region of the lower jaw indicated by the "ellipse", LETA introduces gaps between teeth that were initially proximate. This type of malocclusion poses a challenge

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

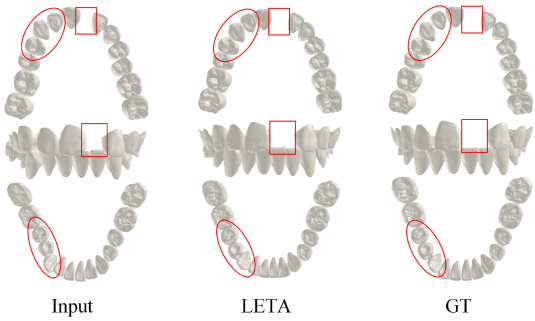


Fig. 13. Generalization on agomphiasis case. "Boxes" indicate the missing tooth. "Ellipses" indicate misaligned teeth.

for LETA due to the significant tooth translation required. Notably, standard orthodontic therapy tends to minimize such tooth translation wherever possible.

- *Openbite:* As shown in Fig. 12(b), where the maxilla and mandible do not touch each other when the mouth is entirely closed. Given that the openbite is relatively rare, LETA cannot acquire enough data to learn during the training phase, which we assume is the main reason for the poor performance.
- *Residual Root:* For some reason, the majority of the crown is destroyed, leaving only the remaining portion of the root, which is named the residual root. As highlighted by "boxes" in Fig. 12(c), it is obvious that LETA, in processing the alignment prediction, moves the position of the residual root towards the direction of the tooth cusp. We surmise that this stems from the residual root's reduced size compared to a complete tooth, leading to more substantial gaps between teeth and subsequently less accurate predictions.

3) Generalization on Agomphiasis: The performance on agomphiasis (missing teeth) case is shown in Fig. 13. Since LETA is designed for inference on a 28-teeth set, in instances of the agomphiasis case, we first fill the empty space using adjacent teeth. For the scenario presented in Fig. 13, the vacant spot left by the missing central incisor is filled with an average of its immediate neighbors: the adjacent central incisor and lateral incisor. Then, this pre-processed data is fed into LETA for pose transformation prediction. Finally, the pose corresponding to the missing tooth is excised from the transformation. As indicated by the "boxes" and "ellipses" in Fig. 13, LETA not only effectively mitigates the impact of the agomphiasis but also adeptly aligns other misaligned teeth.

V. Conclusion

In this paper, we present a novel deep learning based approach for automatic 3D tooth alignment. We introduce a network architecture based on a dual-branch latent encoding, which extracts the shape and context features both from the misaligned and GT aligned teeth with a novel Encoder to guide network learning during the training phase. Both quantitative and qualitative results with a large-scale dataset demonstrate that our method is superior to previous baselines. Moreover, we evaluate the clinical applicability of our method based on a rigorous user study with orthodontists. There are nevertheless some areas to be improved in our method. Particularly, orthodontic rules that are implicitly encoded in the data need to be distilled explicitly so as to impose more constraints on aligned results. A post-refinement to involve explicit orthodontic rules might help alleviate this problem. Besides, it is hard to design clinically applicable treatment plans only with 3D IOS dentitions, as some important anatomical information is missing, such as tooth root, neural tube, and maxillary sinus. The integration of more data modalities, such as CBCT or X-ray, should be considered in the future to help develop more clinically applicable solutions.

References

[1] J. Hao, W. Liao, and Y. Zhang, "Toward clinically applicable 3-dimensional tooth segmentation via deep learning," *J. Dent. Res.*, vol. 101, no. 3, pp. 304–311, 2021.
[2] L. Qiu, C. Ye, P. Chen, Y. Liu, X. Han, and S. Cui, "DArch: Dental arch prior-assisted 3D tooth instance segmentation," 2022, *arXiv:2204.11911*.
[3] H. Xiong et al., "TFormer: 3D tooth segmentation in mesh scans with geometry guided transformer," 2022, *arXiv:2210.16627*.
[4] Z. Cui et al., "TSegNet: An efficient and accurate tooth segmentation network on 3D dental model," *Med. Image Anal.*, vol. 69, 2021, Art. no. 101949.
[5] T.-H. Wu et al., "Two-stage mesh deep learning for automated tooth segmentation and landmark localization on 3D intraoral scans," *IEEE Trans. Med. Imag.*, vol. 41, no. 11, pp. 3158–3166, Nov. 2022.
[6] Z. Cui, C. Li, and W. Wang, "ToothNet: Automatic tooth instance segmentation and identification from cone beam CT images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6361–6370.
[7] M. Chung et al., "Pose-aware instance segmentation framework from cone beam CT images for tooth segmentation," *Comput. Biol. Med.*, vol. 120, 2020, Art. no. 103720.
[8] T. J. Jang, K. C. Kim, H. C. Cho, and J. K. Seo, "A fully automated method for 3D individual tooth identification and segmentation in dental CBCT," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6562–6568, Oct. 2022.
[9] Z. Cui et al., "A fully automatic ai system for tooth and alveolar bone segmentation from cone-beam ct images," *Nature Commun.*, vol. 13, no. 1, 2022, Art. no. 2096.
[10] C. Wang, G. Wei, G. Wei, W. Wang, and Y. Zhou, "Tooth alignment network based on landmark constraints and hierarchical graph structure," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 2, pp. 1457–1469, Feb. 2024.
[11] G. Wei et al., "TANet: Towards fully automatic tooth arrangement," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 481–497.
[12] L. Yang et al., "iOrthoPredictor: Model-guided deep prediction of teeth alignment," *ACM Trans. Graph.*, vol. 39, no. 6, 2020, Art. no. 216.
[13] X. Li et al., "Malocclusion treatment planning via pointnet based spatial transformation network," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, Springer, 2020, pp. 105–114.
[14] S. D. Mangat, "Assessment of malocclusion and orthodontic treatment needs among subjects with dental aesthetic index: A. clinical study," *J. Pharm. bioallied Sci.*, vol. 12, no. Suppl1, 2020, Art. no. S279.
[15] S. S. Alajlan, M. K. Alsaleh, A. F. Alshammari, S. M. Alharbi, A. K. Alshammari, and R. R. Alshammari, "The prevalence of malocclusion and orthodontic treatment need of school children in northern saudi arabia," *J. Orthodontic Sci.*, vol. 8, 2019, Art. no. 10.
[16] L. Dimberg, B. Lennartsson, L. Bondemark, and K. Arnrup, "Oral health-related quality-of-life among children in swedish dental care: The impact from malocclusions or orthodontic treatment need," *Acta Odontologica Scandinavica*, vol. 74, no. 2, pp. 127–133, 2016.
[17] F. Siti, B. S. Tan, and T. Mathew, "Dental esthetic impact of malocclusion and orthodontic treatment need based on self-perception among university students," *Int. J. Dent. Sci. Res.*, vol. 2, no. 2/3, pp. 49–54, 2015.
[18] L. F. Andrews, "The six keys to normal occlusion," *Amer. J. Orthodontists*, vol. 62, no. 3, pp. 296–309, 1972.
[19] Y. Kumar, R. Janardan, and B. Larson, "Automatic virtual alignment of dental arches in orthodontics," *Comput.-Aided Des. Appl.*, vol. 10, no. 3, pp. 371–398, 2013.

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.

[20] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9613–9622.
[21] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[22] A. C. Lin, K.-J. Chang, P.-T. Wu, and P.-E. Yih, "3D cad for design of invisible tooth aligner," in *Proc. IEEE Int. Conf. Mechatron.*, 2005, pp. 647–651.
[23] X. Wang, Z. Li, Y. Ma, and B. Chen, "A method of arrange teeth automatically based on PSO," *Comput. Eng. Appl.*, vol. 48, pp. 211–238, 2012.
[24] Y.-B. Chang, J. J. Xia, J. Gateno, Z. Xiong, X. Zhou, and S. T. Wong, "An automatic and robust algorithm of reestablishment of digital dental occlusion," *IEEE Trans. Med. Imag.*, vol. 29, no. 9, pp. 1652–1663, Sep. 2010.
[25] H. Deng et al., "An automatic approach to reestablish final dental occlusion for 1-piece maxillary orthognathic surgery," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Intervention*, 2019, pp. 345–353.
[26] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.
[27] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 40–49.
[28] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3D object detection from RGB-D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 918–927.
[29] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet : Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.
[30] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. 15th Eur. Conf. Comput. Vis.*, Munich, Germany, 2018, pp. 90–105.
[31] H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6410–6419.
[32] M. Jiang, Y. Wu, and C. Lu, "PointSIFT: A SIFT-like network module for 3D point cloud semantic segmentation," 2018, *arXiv: 1807.00652*.
[33] Z. Zhang, B.-S. Hua, and S.-K. Yeung, "ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1607–1616.
[34] H. Lei, N. Akhtar, and A. Mian, "Spherical kernel for efficient graph convolution on 3D point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3664–3680, Oct. 2021.
[35] 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. pp. 1–12, 2019.
[36] D. Jacob, C. Ming-Wei, L. Kenton, and T. Kristina, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2019.
[37] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10288–10297.
[38] M. Feng, L. Zhang, X. Lin, S. Z. Gilani, and A. Mian, "Point attention network for semantic segmentation of 3D point clouds," *Pattern Recognit.*, vol. 107, 2020, Art. no. 107446.
[39] G. Zhang, Q. Ma, L. Jiao, F. Liu, and Q. Sun, "AttAN: Attention adversarial networks for 3D point cloud semantic segmentation," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2020, pp. 789–796.
[40] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.
[41] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[42] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.
[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[44] M.-H. Guo et al., "PCT: Point cloud transformer," 2021, *arXiv: 2012.09688*.
[45] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Trans. Image Process.*, vol. 6, no. 9, pp. 1305–1315, Sep. 1997.
[46] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1942–1950.
[47] R. Xiong et al., "On layer normalization in the transformer architecture," 2020, *arXiv: 2002.04745*.
[48] 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.*, 2019, pp. 8895–8904.
[49] Z. Wu et al., "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.



Zefeng Shi received the BS degree from the School of Automation, University of Science and Technology of China. He is currently working towrad the master's degree with the State Key Lab of CAD&CG, Zhejiang University. His research interests include image-based modeling and 3D data processing with deep learning.



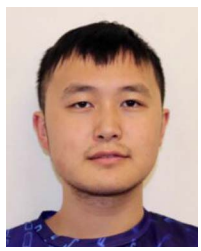
Zijie Meng received the BS degree in computer science and technology from the Taiyuan University of Technology, in 2022. He is currently working toward the MS degree with the Zhejiang University-University of Illinois at Urbana Champaign Institute, Zhejiang University. His research interests include pointcloud and medical image analysis.



Ruizhe Chen is working toward the PhD degree with Zhejiang University. His research interests include domain adaptation, transfer learning and medical image analysis.



Yang Feng received the PhD degree in medical image processing from the University of Edinburgh, in 2015. He worked with the Western General Hospital, Edinburgh and the Asia Research Institution of Philips. He is currently working with Angelalign Inc. in Shanghai as the director of the Software R&D Department. His research interests include pointcloud and graphic processing, CAD automation in orthodontic analysis, diagnosis and treatment planning etc.



Zeyu Zhao received the master's degree in computational science & engineering from Harvard University. He is currently working with Hangzhou ChohoTech Company Ltd. He used to be a senior system software engineer with NVIDIA, focusing on deep learning algorithms development on intelligent video analytics.

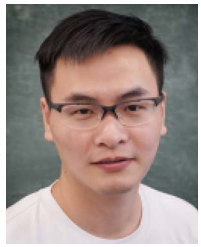
Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.



Jin Hao received the PhD degree from Harvard University, in 2022. He is currently a postdoctoral fellow with Stanford University. His research interests include neuroscience and deep learning-based digital dentistry.



Bing Fang is a head and professor with the Department of Orthodontics, College of Stomatology, Shanghai Ninth People's Hospital, Shanghai Jiao Tong University School of Medicine. She works as a fellow of International College of Dentists (FICD), fellow of Royal College of Surgeons of Edinburgh, United Kingdom, director of the American Edward H. Angle Society of Orthodontist. She is the chairman designate of Chinese Society of Orthodontics(COS), the vice chairman of Chinese Society of Esthetic Dentistry (CSED), and the chairman of Shanghai Society of Orthodontics. She set up Chinese examination center and training center of Royal College of Surgeons of Edinburgh Membership in Orthodontics and opened international orthodontic training courses for countries along the Belt and Road, Shanghai Jiaotong University School of Medicine.



Zuozhu Liu (Member, IEEE) received the BEng degree from Zhejiang University, and the PhD degree from the Singapore University of Technology and Design, in 2019. He is an assistant professor with ZJU-UIUC Institute, Zhejiang University. He is also an affiliated assistant professor with ECE Department, University of Illinois Urbana-Champaign, and the Stomatology Hospital, School of Stomatology, Zhejiang Universtiy School of Medicine. Before that, he was a postdoc research fellow with the Department of Statistics and Applied Probability, National University of Singapore. His research interests are include deep learning, multimodal learning, medical image analysis, and bioinformatics.



Youyi Zheng received the bachelor's and master's degrees in mathematics from Zhejiang University, and the PhD degree in computer science from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, in 2011. He is a researcher with the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University. Prior to that, he spent two years with the School of Information Science and Technology, ShanghaiTech University, where he was also a PI (Researcher). He was a postdoctoral associate with the Department of Computer Science, Yale University from 2013 to 2015, and a postdoctoral fellow with KAUST from 2011 to 2013. His research interests include geometric modeling, imaging, and human-computer interaction.

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:05:14 UTC from IEEE Xplore. Restrictions apply.