3D Deep Learning for Orthodontic Attachment Strategy Based on PointNet++

Shiyi Cao, Rong Zhou
Department of computer and science, SJTU

Abstract—The combination of medical science with artificial intelligence has becoming an emerging topic. The Methods of Machine learning can significantly contribute to the strategies in medical treatment. In the orthodontic process, there are many decisions to be made, such as when to add an attachment on which tooth, which type of attachment should be added. By means of deep learning, we can extract features from the model of teeth and use attachment labels to train the network. In this way we can help dentists making decisions in different scenarios.

Index Terms—Orthodontics, Point Cloud, Deep Learning

1 Introduction

DURING the orthodontic process, some attachment would be added to teeth to provide a force point on the surface. There are 4 elements in the attachment decision: type, orientation, surface and start stage. For each element, several choices are offered, as shown in Table. 1. With deep learning techniques, we are able to predict whether an attachment should be added to a tooth, how many attachments should be added and what kind of attachments should be chosen. Typically, given a teeth model, the prediction task can be done in 2 steps: feature learning and classification (or segmentation).

TABLE 1: Elements in the Attachment Decision

Type	Ellipsoidal, Optimized Extrusion Optimized Rotation, Rectangular
Orientation	Vertical, Horizontal
Surface	Buccal, Lingual
Start Stage	$0, 1, \ldots, 99$

The challenges for 3D deep learning lie in the processing of the point cloud, which is a collection of points captured by 3D scanners. Such data are invariant to permutations of its members [1]. Moreover, the density and other attributed are not uniform across different locations [1]. Due to the irregular format of point cloud [2], traditional methods tend to transform these data to regular 3D voxel grids or collection of images to feed them into convolutional deep networks. However, such methods may suffer from issues like introducing quantization artifacts that can obscure natural invariances of the data [2]. Therefore, like many researchers nowadays, we decide to simply use point cloud as input. The feature learning process is done leveraging PointNet++ [1], a novel 3D deep learning architecture that can learn the point cloud feature efficiently capturing features in both local and global structure.

Another challenge in the task of predicting the attachment strategy is that it is a multi-label classification problem. Take one teeth model as example, there are 32 teeth in one teeth model with at most 3 attachment added on each tooth and for every attachment, 4 elements should be decided. As a result, the prediction results for one teeth model will contain so much information that it can be very complex.

To address all of these problems, we propose 2 models to predict the attachment strategy in orthodontic process. In feature learning stage, we use strutures introduced in [1]. In classification stage, we try two different methods and test both of our models on real-world datasets.

2 Feature Learning

In this part we introduce the feature learning stage in PointNet++ model to extract features from point cloud. This model can capture local structures induced by the metric space points live in and recognize fine-grained patterns as well as generalizability to complex scenes [1].

The feature learning stage is a hierarchical version of PointNet [2]. Each layer has three sub stages: sampling, grouping, and PointNeting. In the first stage, they select centroids and in the second stage, they take their surrounding neighboring points (within a given radius) to create multiple sub-point clouds. Then they feed them to a PointNet and get a higher dimensional representation of these sub-point clouds.They repeat the process (sample centroids, find their neighbors and Pointnet on their higher order representation to get an even higher one) 3 times.

By exploiting metric space distances, the network is able to learn local features with increasing contextual scales.

In the single structure of PointNet, each point was projected to a 1024 dimension space. The order problem is solved by using a symmetric function (max-pool) over the points. This yielded a 1 x 1024 global feature for every point cloud which is fed into a nonlinear classifier. The rotation problem is settled by using a mini-network called T-net. It learns a transformation matrix over the points (3 x 3) and over mid-level features (64 x 64). Calling it mini is a bit misleading since it is actually about the size of the main network. In addition, because of the large increase in the number of parameters they introduced a loss term to

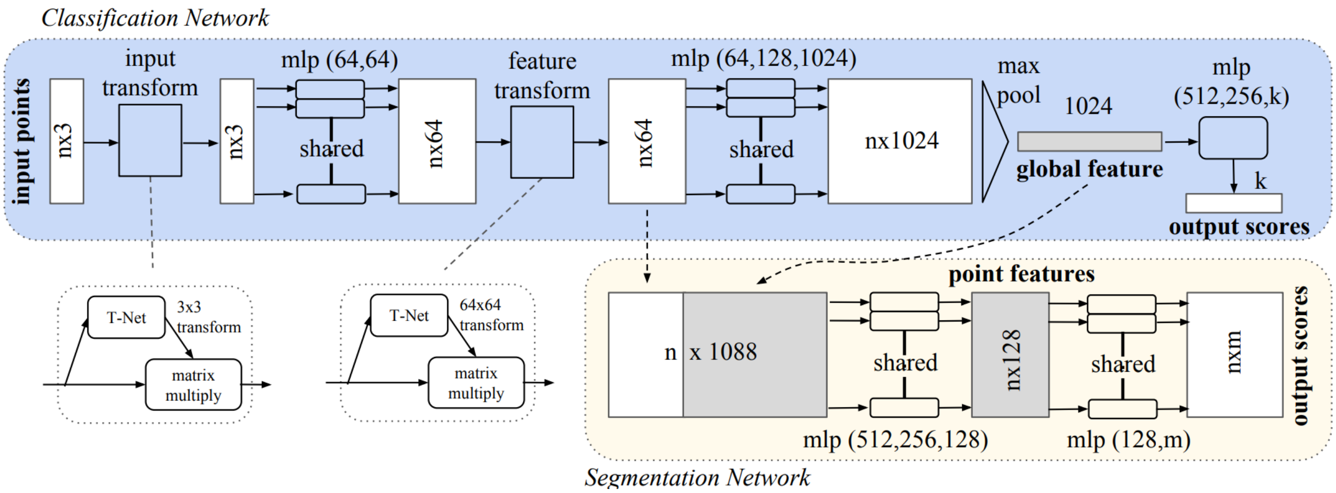


Fig. 1: PointNet Architecture. The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores fork classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. mlp stands for multi-layer perceptron, number sin bracket are layer sizes. Batch norm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification ne.

constraint the 64 x 64 matrix to be close to orthogonal. The network structure of $PointNet$ is shown in Fig. 1

3 Classification

3.1 Parallel Classification

There are four labels in the orthodontic process: attachment_id, type, orientation, surface, and start stage. By observation, we find that each tooth has at most 3 attachments. Regarding that every object has 32 teeth T, we use a $[32 * 3]$ vector IV to represent each label L. We also number the categories in each label, for instance, number "Buccal", "Lingual" as "1", "2" in the surface label. In addition, "0" means one tooth doesn't has any attachment on it.

$$IV = [T1_{L1}, T1_{L2}, T1_{L3}, T2_{L1}, \cdots, T32_{L3}] \qquad (1)$$

We feed the point data of 10000 dimensions in each object to the feature learning stage and then get the feature vectors. We process the feature vectors by two fully connected layers and two dropout layers in order to avoid overfitting. The first layer is of 12800 dimensions and the second layer is of 6400 dimensions. The drop rate is set to 0.5. In this section we assume the five kinds of labels are independent, hence we use five parallel fully connected layers to make the prediction for each label instead of a single layer in the $Pointnet++$ model. A $[96 * n]$ vector is used to represent prediction in the form of one-hot where n is the categories of each label. The network structure is shown in Fig. 2.

Then we apply cross entropy for calculating the loss of each label separately and then add them together as the loss of the whole model to do backpropagation.

$$Loss = \sum_{i=1}^{4} Loss_{label_i} \qquad (2)$$

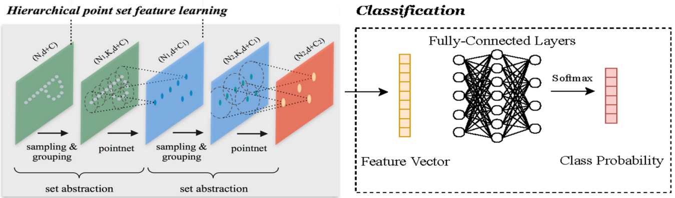


Fig. 2: Fully connected structure.

3.2 Point2Seq

3.2.1 Important Findings

To further improve our model performance, we make analysis on the correlation between different kinds of factors based on the attachment data of 75 patients, as shown in Fig. 3. It is estimated that over 80% of the attachments belong to the 9 classes shown in Fig. 3 and in Table. 2. The attachments are classified into 49 classes. Class 1 is specially set for "no attachment". The other 48 classes each presents a combination of the choices in those 4 factors. Notably, we simplify the choices for *Start Stage*: 0, 3 or more than 3, since we have found that 90% of the attachement strategies start at stage 0 or 3. Another intuition is that, attachment strategy for one tooth may have strong impact on the other teeth: to openbite malocclusion a pair of attachments will be attached on the corresponding lower tooth and upper tooth. We then try another classification method that considers the inner correlation among the 4 factors as well as the interdependence among teeth.

3.2.2 Encoder-Decoder Structure

Point2Seq is designed based on encoder-decoder structure, which generates sequential classification results of attachment strategy for each tooth. The overview model of *Point2Seq* is demonstrated in Fig. 5. Encoder-decoder struc-

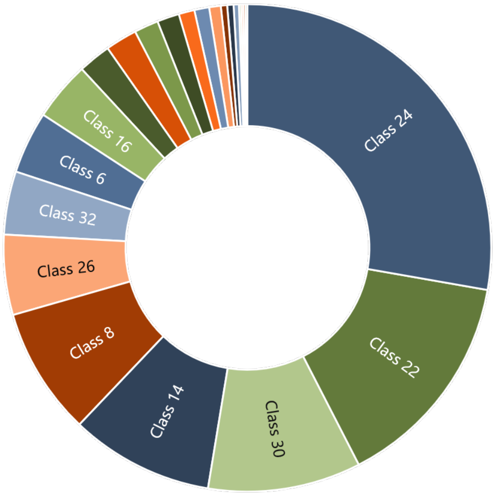


Fig. 3: Occurrence frequency of attachment classes.

TABLE 2: Top 5 most popular attachment classes.

Top 5	Account for 61%
Class 24	Stage 3, Optimized Rotation, Horizontal, Buccal
Class 22	Stage 3, Optimized Rotation, Vertical, Buccal
Class 30	Stage 3, Rectangular, Vertical, Buccal
Class 14	Stage 0, Rectangular, Vertical, Buccal
Class 8	Stage 0, Optimized Rotation, Horizontal, Buccal

ture [3] is the state-of-the-art solution to machine translation problem. We try in our work to consider the task of generating attachment strategy for each tooth as a translation problem, which means we "translate" the teeth feature vector into proper attachment strategy.

Decoder: The decoder is a Long Short-term Memory Network (LSTM), which can catch the long-term and short-term correlation among different teeth. We assume that there are at most 3 attachments on each tooth. Then for each teeth model, given the feature vector otained in the feature learnig process, the output of the decoder can be presented as Eq. (3), where T_i^j is a one-hot vector indicating which class the j-th attachment of the i-th tooth belong to.

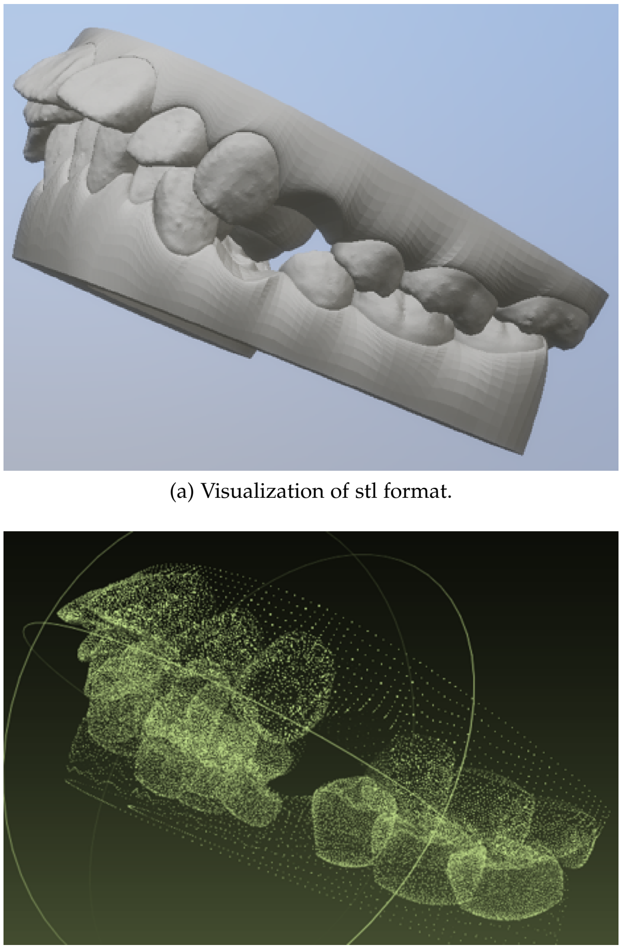
$$Decoder_{out} = \{T_1^1, T_1^2, T_1^3, \ldots, T_{32}^1, T_{32}^2, T_{32}^3\} \tag{3}$$

In decoder, h_t is decided not only by h_{t-1} and the input v, but also the output y_{t-1} at time $t-1$, as shown in Eq .(4)

$$h(t) = f(h_{t-1}, y_{t-1}, v) \tag{4}$$

Overall Loss Function: The loss function computes the cross enthrophy of the class prediction for each teeth model:

$$L = -\sum_t y_t \log p(y_t|x) \tag{5}$$



(a) Visualization of stl format.

(b) Visualization of point data.

Fig. 4: Data preprocessing

4 Evaluation

4.1 Data Preprocessing

The source data is in stl format and we use meshlab to change raw data into xyz format. We also use basic sampling method to reduce the number of pointdata from about 40000 to precisely 10000. Each point of the tooth is represented by a 3 tuple (x, y, z). The visualizations of the raw data and the point data are as follows in Fig. 4.

4.2 Settings

We implement our models with Tensorflow and test them on a real-world dataset that contains 75 original teeth models before orthodontic process in stl format and the corresponding attachment strategy given by professional dentists. We divide the dataset into two part: 50 for training and 25 for testing. The performance of the models are evaluated based on the accuracy of their predictions. The accuracy for one prediction is defined as the proportion of the attachment strategies that are predicted right. Here we denote the first model as *ParaC* and the second model as *Point2Seq*.

4.3 Loss

Figure. 6 illustrates the change of the loss during training and testing process. Both the two model reaches a relatively

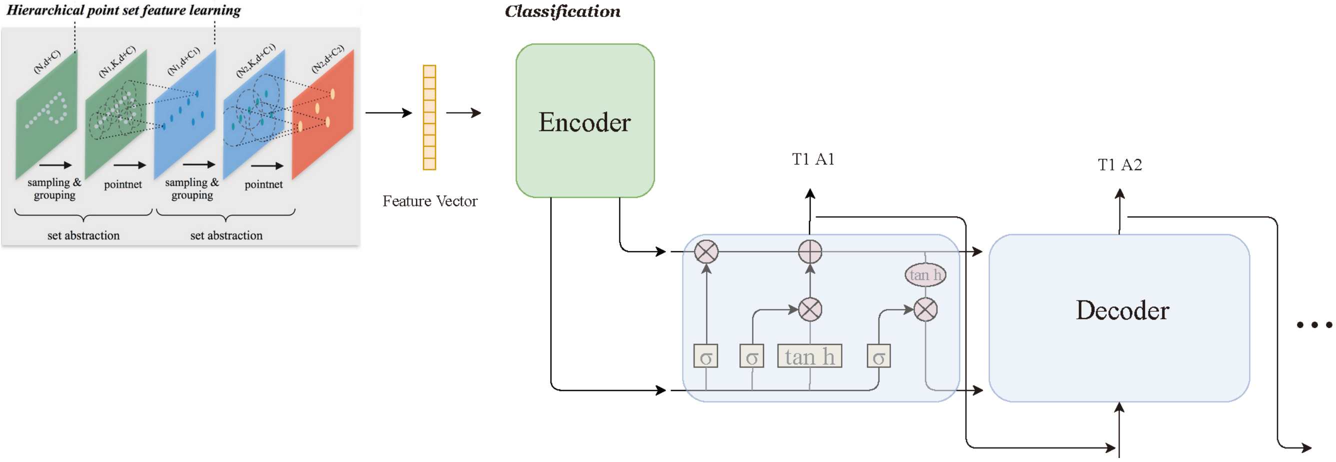
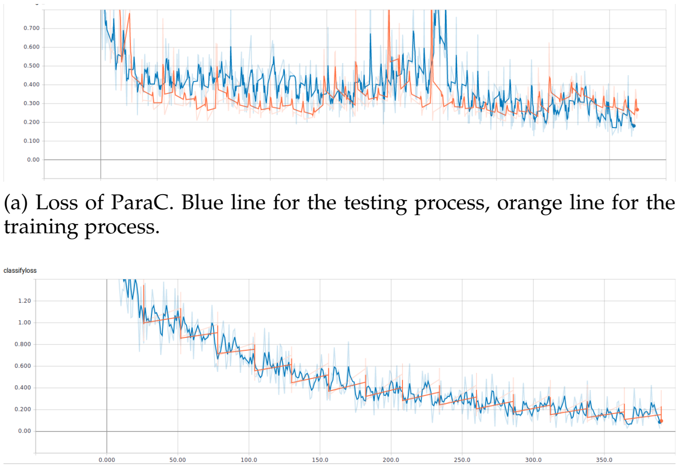


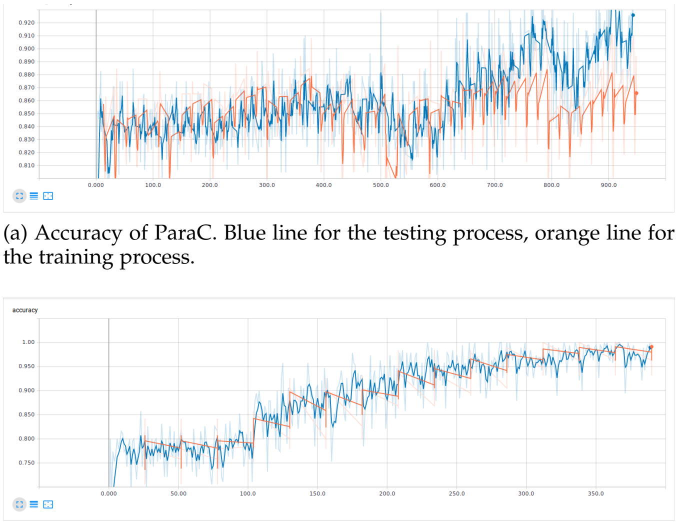
Fig. 5: A running example of Point2Seq. The feature vector learned by PointNet++ is encoded and put to the decoder. The decoder consists of several LSTM cells that exploit the interdependence among teeth. The decoder sequentially outputs vectors representing the probability of classes that the corresponding teeth belong to.



(a) Loss of ParaC. Blue line for the testing process, orange line for the training process.

(b) Loss of Point2Seq. Orange line for the testing process, blue line for the training process.

Fig. 6: Loss of the models.



(a) Accuracy of ParaC. Blue line for the testing process, orange line for the training process.

(b) Accuracy of Point2Seq. Orange line for the testing process, blue line for the training process.

Fig. 7: Accuracy of the models.

low loss degree. ParaC makes it at around 0.17 while Pointn2Seq can reach around 0.1. We can see from the loss of ParaC that there are sometimes sudden changes in the loss value, which we attribute to the randomness of the parallel training methods. Since in ParaC we train on each element independently, we assume that the choice made concerning one element of the attachment strategy is only dependent on the teeth model. However, the fact is other aspects of the attachment strategy may have equal influence on it. Thus Point2Seq shows a more stable performance.

4.4 Accuracy

The accuracy of the two models are shown in Fig. 7, the blue line represents the testing process while the orange line representing the training process. In our experiments, the accuracy of *ParaC* reaches around 0.88 while that of *Point2Seq* reaches around 0.97, which shows that the consideration of the correlation among different elements is necessary, since there are some regular patterns in the attachment strategy regardless of the feature of the teeth model. Moreover, the sequential correlations among attachment strategy for different teeth learned by the network may make up for the feature learning process that sometimes fails to extract a proper feature.

5 Conclusion and Future Work

We construct two models based on PointNet++, one is based on the independency of the four labels while the other is based on the relation between the four labels. Statistic analysis of the raw sample attachment labels helps to better our model design. A novel Point2Seq learning model leveraging Encoder-Decoder structure is introduced, which can generate sequential predictions exploiting the correlation among teeth. Hopefully, this idea can be helpful in other similar scenarios.

Nevertheless, there are some limitations in our model construction and feature learning. First, the sample data is inadequate, which may increase the risk of overfitting. Second, we are utterly ignorant of stomatology and we use pure machine learning to predict the attachment labels. If we can combine medical theory with deep learning, the predictions of attachment labels will have significant improvements. Third, we just extract the features from dental crown. In future work, we should also consider extract features from tooth neck and root and combine them together.

Acknowledgment

In our work, Rong Zhou is responsible for the ParaC model and data preprocessing. Shiyi Cao proposes the Point2Seq model, makes sampling on the point cloud data and conducts several data analysis on the correlation among teeth and elements.

We would like to thank our TA and other classmates who offered us much help during this project.

References

[1] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *Annual Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 5105–5114.

[2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2017, pp. 77–85.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Annual Conference on Neural Information Processing Systems (NIPS)*, 2014, pp. 3104–3112.