Neural Orthodontic Staging: Predicting Teeth Movements with a Transformer

Jiayue Ma, Jianwen Lou†, Borong Jiang, Hengyi Ye, Wenke Yu, Xiang Chen, Kun Zhou, Youyi Zheng†

Abstract—We present a novel learning-based method for predicting tooth movements in orthodontic treatment path planning (orthodontic staging). Recognizing the multi-solution nature of orthodontic staging, our approach involves generating the staging sequence progressively with a dedicated Transformer model. This model predicts teeth movements within a predefined number of steps (e.g., 10 or 20), targeting alignment in problematic dentition. The Transformer refines its predictions iteratively, building on previous outcomes until reaching a state that aligns with the target within an acceptable distance. This mirrors real-life scenarios where orthodontists dynamically adjust staging plans based on treatment outcomes. Our Transformer model is tailored to incorporate spatial and temporal attentions, addressing inter-tooth and inter-step interactions, respectively. These attentions are further refined with relative positional encoding. Recognizing the significant influence of tooth shape on the alignment process, we propose integrating a tooth-wise shape encoder to extract morphological features from the 3D teeth point cloud. These features are then fused into the Transformer, facilitating the capture of inter-tooth dynamics during staging, in collaboration with spatial attention. We validate the proposed method on a large-scale dataset that contains 10K real-life orthodontic cases. The results show that our method outperforms the state-of-the-art, and orthodontists favor its predictions.

Index Terms—orthodontic staging, transformers, shape encoding

1 Introduction

Orthodontics aims at aligning our bite and straightening our teeth. However, the treatment planning of orthodontics is non-trivial due to complex dental conditions such as crooked, gapped, or overlapping teeth. Moreover, factors like dental health, arch width, and cranial condition pose additional challenges to orthodontic decision-making. As a remedy, staging breaks down an orthodontic treatment plan into sequential steps. The teeth are moved mildly toward the aligned target pose within each step. This strategy allows orthodontists to monitor how the treatment progresses and make timely adjustments for a desired outcome.

Despite its importance, orthodontic staging is mainly performed by hand in clinical practice, which is time-consuming and labor-intensive. An automated staging method is thus in demand. Previous studies typically treat staging as an optimization problem by integrating medical rules (e.g., no interdental collision) into the objective function, then solve it with traditional algorithms such as genetic algorithm [1], A* algorithm [2] and optimized artificial bee colony algorithm [3]. Unfortunately, these approaches are prone to local optimum and inept in addressing intricate medical rules (e.g., facial aesthetics and orthodontic order) that are difficult to formulate mathematically. Such inherent defects significantly limit the approach's overall staging performance while impeding it from being applied to real-life clinical cases. As an alternative, deep learning-based approaches have pushed the frontiers of many fields, and orthodontics is no exception (e.g., automated tooth alignment). However, it remains under-explored for automated orthodontic staging. The following challenges hinder the progress. First, misaligned teeth are in various forms, resulting in a complex distribution of staging sequences that is hard to model. Second, the poses of neighboring teeth are dependent on each other, which must be carefully treated during staging. Third, staging is prone to accumulating errors step-by-step due to its sequential nature. Fourth, orthodontic staging inherently accommodates multiple viable solutions. In practice, orthodontists adapt the staging plan based on observed treatment outcomes, resulting in diverse pathways of varying lengths, even from similar initial states. This variability makes it challenging to predict the precise number of staging steps in advance. Many existing methods assume a fixed step count and use interpolation to generate intermediate poses simultaneously, overlooking the intrinsic multi-solution nature of orthodontic staging.

In this paper, rather than generating a complete staging sequence concurrently, we adopt an iterative approach to mimic real-life scenarios. We propose recasting orthodontic staging into a sequence-to-sequence prediction framework, leveraging Transformers as the backbone. Transformers are renowned for their proficiency in managing sequential data and capturing intricate data dependencies through the attention mechanism. In orthodontic staging, where there are complex interactions between teeth and sequential treatment steps, the attention mechanism proves invaluable. This capability positions Transformers as a highly suitable choice for iterative predicting in orthodontic staging. Initially, we train a Transformer to forecast the movement tendencies of teeth from any problematic state towards alignment within a limited number of steps (e.g., 10 or 20 steps). Subsequently, we iterate the Transformer's predictions until they converge to an acceptable deviation from the target aligned dental state. The final state of the last prediction round serves as the starting point for the Transformer in the next iteration. The ultimate staging sequence is obtained by concatenating these short-term predictions from the Transformer. Our Transformer incorporates

†*corresponding author*

- *J. Ma, B. Jiang, H. Ye, X. Chen, K. Zhou, and Y. Zheng are with the State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China, 310058.*
  J. Lou is with the School of Software Technology, Zhejiang University.
  W. Yu is with the Chengxi Branch of Hangzhou Stomatology Hospital.
  E-mail: youyizheng@zju.edu.cn, jianwen.lou@zju.edu.cn

spatial attention to model inter-tooth relationships and temporal attention to grasp the progression across steps. To enhance the Transformer's comprehension of both the spatial arrangement of teeth and their temporal dynamics during orthodontic staging, we augment it with dual unlearned relative positional encodings. To accurately represent tooth morphology, we employ a tooth-wise shape encoder to extract shape features from 3D teeth point clouds as shape codes, which are then integrated into the Transformer. The collaboration between shape codes and spatial attention significantly enhances the network's capacity to simulate dynamic relationships among teeth. We verify the effectiveness of the proposed iterative Transformer-based approach on a dataset comprising 10K real-world orthodontic staging cases, each providing a sequence of 3D teeth point clouds captured along the staging process with a high-precision oral scanner and the corresponding pose labels. Our method achieves state-of-the-art performance in neural orthodontic staging. The results are quantitatively sound and favored by orthodontists.

In summary, the main contributions of our work are:

- We present the first Transformer-based method for orthodontic staging, featuring an iterative framework that leverages the Transformer for short-term predictions.
- We integrate spatial attention to capture inter-tooth correlations and temporal attention to model inter-step relationships into our Transformer, equipped with unlearned relative positional encodings. We also incorporate shape codes extracted from a tooth-wise shape encoder to enrich the network's comprehension of tooth morphology.

2 Related Work

2.1 Automated Orthodontic Staging

Orthodontic staging, or orthodontic path planning, divides the movement of teeth into a series of stages based on the initial and target poses of teeth such that an aligner can faithfully align the teeth by the stages while obeying the biomechanics. Existing methods for automatic orthodontic path planning are primarily optimization-based. Li et al. [1] use a genetic algorithm to find the optimal path for teeth movement, with the objective function defined as the weighted sum of movement distance, rotation angle, and motion constraints. However, genetic algorithms tend to premature convergence and may get stuck in local optima. Li et al. later use A* algorithm [2] and improve artificial bee colony algorithms [3] to address orthodontic staging. To accelerate the running speed, Xu et al. [4] apply the particle swarm optimization algorithm to optimize teeth movement paths and use oriented bounding boxes for collision detection. However, their approach, which treats all teeth as a single particle, does not consider the varying speeds of different teeth. To address this issue, Ma et al. [5] assign different inertia parameters to particles to distinguish different teeth. Despite its simplicity, particle swarm optimization performs suboptimal when dealing with high-dimensional problems such as orthodontic staging and is prone to local optima. A recent optimization method [6] improves the convergence factor and position update strategy of the grey wolf algorithm (IGWO), encoding the movements of all teeth across multiple stages using a single grey wolf entity and initiating through interpolation. These optimization-based methods mainly focus on minimizing the total movement distance of teeth while neglecting the movement order. Their case-specific design and the lack of diverse data samples significantly limit their capacity to handle complex cases. In this paper, we propose a data-driven method that employs a Transformer-based neural network to learn the artificial staging patterns from a large-scale dataset of various clinical staging cases. Leveraging real clinical insights enables our approach to address complex orthodontic challenges more effectively. A concurrent work [7] proposes a collaborative tooth motion diffusion model that redefines orthodontic tooth motion planning as a diffusion process, integrating inter-tooth and occlusal constraints through graph structures and novel loss functions to enhance the learning of multi-tooth motion distributions. However, this method requires predefining the number of staging steps, limited by the prediction length and computation time.

2.2 Learning on 3D Point Cloud

Numerous studies have delved into deep learning techniques in point cloud learning with applications including classification, segmentation, object detection, tracking, registration, and completion. Our research targets the specific challenge of representing tooth shapes using point clouds to assist in orthodontic staging generation. Key efforts in point cloud learning include PointNet [8] and its improved version, PointNet++ [9], which become prominent for their ability to address the point cloud disorder problem. PointNet [8] introduces a transformation network and a symmetric function to ensure permutation invariance for unordered points. PointNet++ [9] enhances it by incorporating a hierarchical structure to extract local features at multiple scales. Due to CNN's ability to share weights, some works [10], [11], [12] adopt it for point cloud feature learning. Graph-based models like Dynamic Graph CNN [13] focus on capturing relationships between points to improve feature learning. Unsupervised learning methods, particularly autoencoders, have been explored for learning point cloud representations [14], [15], [16]. FoldingNet [14] introduces a peculiar decoder design to simulate a 2D-to-3D mapping based on an autoencoder, proving useful in point cloud completion tasks [17], [18]. TopNet [19] proposes a novel decoder to generate structured point clouds without assuming any particular structure or topology. Led by PCT [20] and Point Transformer [21], a substantial body of transformer-based works have emerged in the field of point cloud processing [18], [22], [23], [24]. These methods have shown promising performance on publicly available datasets, but the high dimensional encoder output features are typically bloated for the implicit representation of tooth shapes. Since the teeth are similar, we set a compact encoding size.

2.3 Sequence-to-sequence Prediction

Sequence-to-sequence(Seq2Seq) models, originally designed for tasks like machine translation, have been progressively applied to a range of applications, including text summarization, speech recognition, and video synthesis. These models are adept at transforming input sequences into output sequences, making the Seq2Seq framework well-suited for predicting the stages of orthodontic treatment plans.

Seq2Seq prediction is fundamentally based on the encoder-decoder architecture. The encoder processes the input sequence, capturing its essential information into a fixed-length vector and the decoder then generates the output sequence from this vector. Early Seq2Seq models [25], [26]relied on recurrent neural networks (RNNs) and their variants, such as Long Short-Term

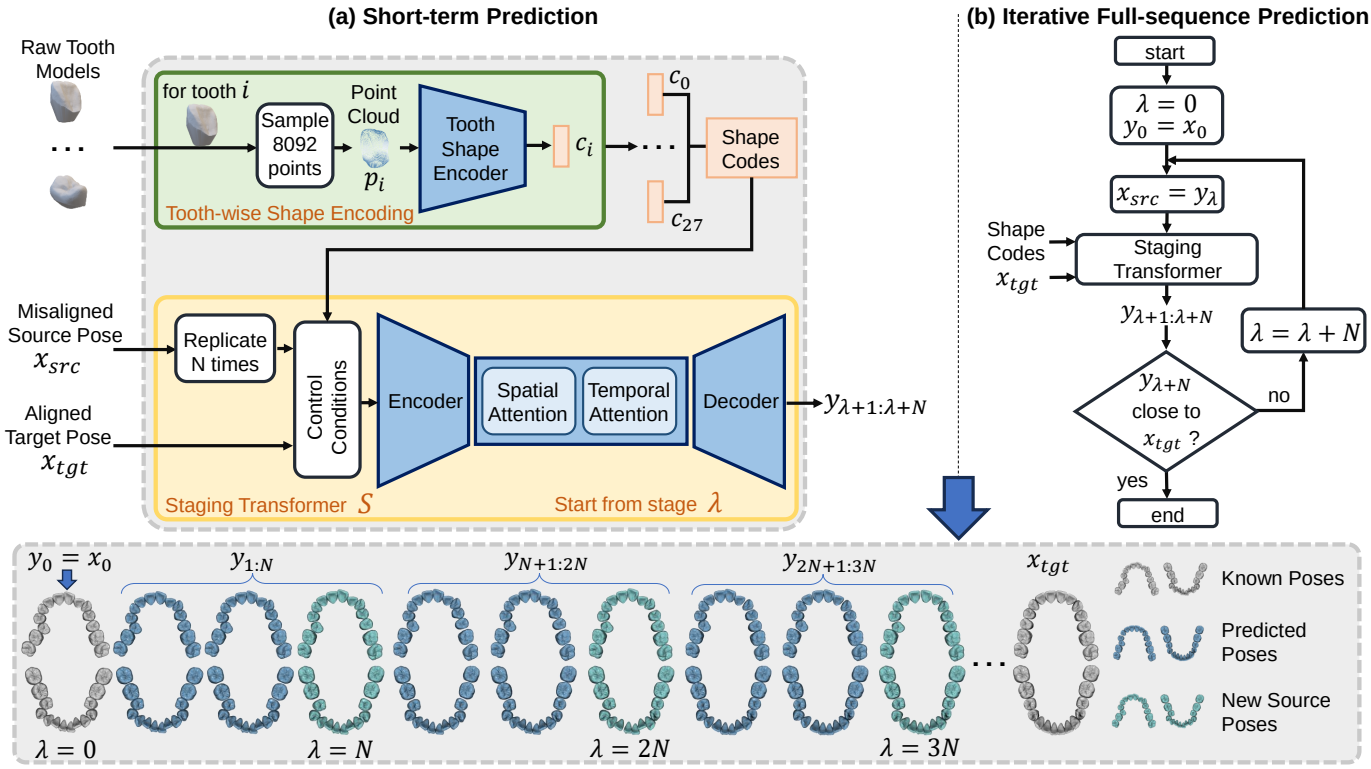


Fig. 1. The overall pipeline of our method. Given the raw tooth models, the misaligned source pose x_{src}, and the aligned target pose x_{tgt}, our network performs short-term predictions as depicted in (a). Within the tooth-wise shape encoding module, we sample point clouds p_i for each tooth model i, then employ the tooth shape encoder to extract shape features c_i. The shape features of all teeth constitute the shape codes. The shape codes, together with the replicated source pose and target pose, are fed into the staging transformer module to forecast the next N steps. To generate a full staging sequence, we iteratively apply the network to estimate intermediate results as shown in (b). Each prediction round's endpoint serves as the next round's starting point, continuing until the predicted poses closely approximate the aligned target pose.

Memory (LSTM) networks, to manage sequence data. However, challenges like vanishing gradients, accumulating errors and computational inefficiency often hamper these recurrent models' ability to forecast over a long range. Convolutional networks enable parallel processing of sequence elements, unlike RNNs, which rely on sequential hidden states that hinder parallel computations within a sequence. Gehring et al. [27] hence proposes an entirely convolutional architecture for quicker training and easier optimization.

The Transformer model proposed by Vaswani et al. [28] marks a paradigm shift in Seq2Seq prediction. Transformers eschew recurrence in favor of self-attention mechanisms, allowing the model to weigh the importance of different parts of the input sequence directly. This architectural innovation bolsters the model's proficiency in capturing long-term dependencies and markedly enhances training efficiency. The development of pre-trained models further demonstrates the Transformer's effectiveness across various NLP tasks. Notably, BERT (Bidirectional Encoder Representations from Transformers) [29], which learns contexts from both sides of the text, establishes a new benchmark in language understanding. Moreover, Radford et al. [30] propose GPT-2 trained on extensive textual corpora to predict texts, pushing the boundaries of generative text models. Beyond text, the Transformer has found applications in speech recognition [31], [32], music generation [33], [34], time series forecasting [35], [36], video prediction [37], [38], and video inpainting [39], [40], showcasing its versatility for processing various input sequence types. The data structure of orthodontic staging is akin to that of videos, inspiring us to learn spatial and temporal dynamics via Transformers. We frame orthodontic staging as a Seq2Seq prediction task and propose a data-driven strategy armed with Transformers. This method excels in modeling long-term dependencies, allowing for effective capture of inter-tooth and inter-step relationships.

3 METHOD

3.1 Overview

Staging in orthodontics involves planning the trajectory of tooth movement, dividing the orthodontic treatment into a sequence of steps wherein teeth progressively shift toward an organized alignment. Existing methods predetermined the number of steps and estimated the steps between the given source and target dentitions to address the staging challenge [1], [2], [4], [5], [6], [7], [41]. This strategy significantly restricts the solution space. In contrast, real-world orthodontic practice necessitates regular treatment adjustments by dentists based on patient responses and orthodontic progress, implying that even with similar initial and final dental states, the actual movement path and step count can vary. The inherent multi-solution nature of staging demands a dynamic approach. With this insight, we posit that the crux of staging lies in short-term forecasting from any problematic dentition.

The new problem is formulated as follows: with the goal of achieving an aligned dentition x_{tgt}, the objective is to predict the moving pathway of the teeth $\{y_{\lambda+1}, y_{\lambda+2}, ..., y_{\lambda+N}\}$ over the next N steps, starting from a misaligned initial dentition $x_{src} = y_\lambda$

along with its corresponding 3D teeth point cloud. In the context of a dentition $y_t \in \mathbb{R}^{28\times 9}$, it refers to the 3D positions and 6D rotations of 28 teeth relative to a predefined reference at step t. We propose a novel Transformer structure to tackle the problem. This Transformer incorporates a custom attention mechanism, utilizing spatial attention for inter-tooth correlation and temporal attention for inter-step correlation. Shape features for each tooth are extracted using a tooth-wise shape encoder, collected to form a shape code that, in collaboration with spatial attention, captures dynamic inter-tooth relationships. With this Transformer-based prediction model, we iteratively finalize the staging sequence updating $\lambda = \lambda + N$ and using intermediate outcomes, which mirrors real-world scenarios with dynamic adjustments to the pathway.

3.2 Tooth-wise Shape Encoding

Dental conditions are affected by both tooth poses and shapes. Although the spatial relationship between teeth may vary during the staging procedure, tooth shapes remain constant. To address this, we propose to independently extract shape features for each tooth using an identical tooth shape encoder. These features as shape codes are transformed in the control condition module, collaborating with the spatial attention mechanism (refer to Sec 3.3.2 for details) in the transformer to capture dynamic inter-tooth relationships during staging.

For a tooth i, before encoding its tooth shape, we extract a point cloud $p_i \in \mathbb{R}^{8092\times 3}$ by sampling 8092 points over the raw 3D tooth scan, and align the point cloud with a unified reference coordinate system. We employ the transformer-based encoding mechanism outlined in [18] as the shape encoder $\mathbf{G}$, taking 8092 points p_i as input. It produces a 108-dimensional vector $c_i \in \mathbb{R}^{108}$, which serves as the shape code of the respective tooth i. In cases where teeth are extracted before the orthodontic intervention, resulting in missing models and poses, we set the shape codes for these teeth to a zero vector $\mathbf{0}^{108}$. We retrain our encoder and its paired decoder [18] with our tooth data on a point cloud completion task, aimed at reconstructing full point clouds from partial inputs. We optimize the encoder-decoder networks by minimizing the chamfer distance between the reconstructed point clouds and their original counterparts. During staging generation, we discard the decoder and solely utilize the pre-trained encoder with its parameters fixed, where the encoder's input is the complete original point cloud. The encoding process can be formulated as below:

$$c_i = \mathbf{G}(p_i) \tag{1}$$

For integration into the generative staging transformer, we use the 28 vectors $c_{0:27}$ as a control condition and prepend them to the existing dental state sequence. See Sec 3.3.1 for specific operations.

3.3 Staging Transformer

Leveraging the transformer's proven success in natural language processing and its capability to handle long-range dependencies, we utilize it as the backbone of our staging generation network S. This network, depicted in Figure 2, adheres to an encoder-transformer-decoder architecture. It processes an arbitrary misaligned dental state $x_{src} = y_\lambda$, a target aligned state x_{tgt}, and the shape codes $c_{0:27}$, subsequently producing the future N steps $y_{\lambda+1:\lambda+N}$. At each step t, the state x_t or y_t comprises the global

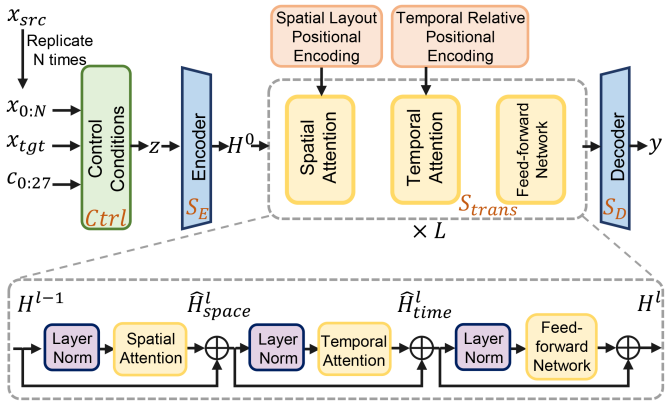


Fig. 2. The overall structure of the Transformer-based staging generation network.

positions $p_t \in \mathbb{R}^{28\times 3}$ and 6D rotations [42] $r_t \in \mathbb{R}^{28\times 6}$ of 28 teeth. To fit the sequence-to-sequence transformer model, we replicate x_{src} for N times and replace the unknown values in the next N steps denoted as $\{x_n = x_{src}\}_{n=0}^{N}$. We model the staging generation as:

$$y = S(x_{0:N}, x_{tgt}, c_{0:27}) \tag{2}$$

$$S(\cdot) = S_D(S_{trans}(S_E(\text{Ctrl}(\cdot)), L)) \tag{3}$$

where $\text{Ctrl}(\cdot)$ denotes the Control Condition module, which integrates dental states $x_{0:N}$ with shape codes and target state as z. The encoder S_E, transformer layers S_{trans} and decoder S_D correspond to three sub-networks within the staging transformer, respectively.

The encoder S_E transforms the dental states z into D-dimensional latent embeddings $H^0 = S_E(z)$ through a fully connected network, comprising linear and activation layers.

Our transformer layers S_{trans} follow the canonical transformer architecture, featuring Multi-Head Self-Attention (MHSA), Feed-Forward Networks (FFN), and residual connections. We distinguish our transformer with three key modifications. First, we incorporate shape codes and target state into our transformer as control conditions. Second, we sequentially use spatial self-attention and temporal self-attention modules within a single layer. Third, we add relative positional encoding to the attention scores as a bias matrix in both spatial and temporal attention blocks (Spatial Layout Positional Encoding and Temporal Relative Positional Encoding), instead of using absolute positional encoding before entering the transformer layer. The processing flow in S_{trans} can be formulated as:

$$\hat{H}^l_{space} = H^{l-1} + MHSA_{space}(\text{LayerNorm}(H^{l-1})) \tag{4}$$

$$\hat{H}^l_{time} = \hat{H}^l_{space} + MHSA_{time}(\text{LayerNorm}(\hat{H}^l_{space})) \tag{5}$$

$$H^l = \hat{H}^l_{time} + PFFN(\text{LayerNorm}(\hat{H}^l_{time})) \tag{6}$$

where H^l represents the output of layer-l ($l = 1, \ldots, L$).

The decoder S_D, structurally similar to the encoder, is a fully connected network that transforms the D-dimensional output H^L of the transformer layers into the predicted staging sequence y.

We optimize the Staging Transformer S by minimizing the per-step reconstruction loss L_{rec} and the smoothness loss L_{smooth} over the estimated steps. The overall loss functions are calculated as follows:

$$L_{total} = L_{rec} + L_{smooth} \tag{7}$$

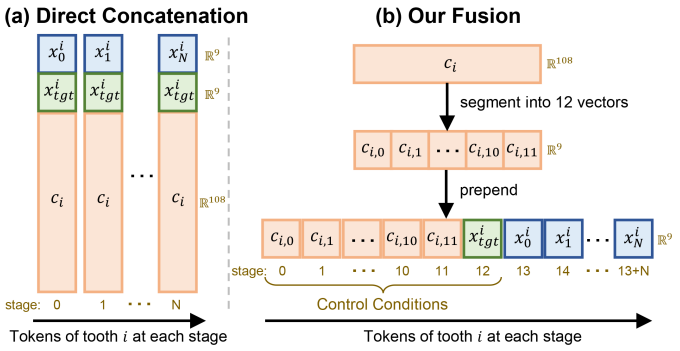


Fig. 3. We demonstrate two strategies for integrating control conditions into tooth poses: direct concatenation (a) and our fusion strategy (b).

$$L_{rec} = \frac{a_{pos}}{N}\sum_{t=1}^{N}\|p_t - \hat{p}_t\|_1 + \frac{a_{rot}}{N}\sum_{t=1}^{N}\|r_t - \hat{r}_t\|_1 \tag{8}$$

$$L_{smooth} = \frac{b_{pos}}{N}\sum_{t=1}^{N}\|\hat{p}_t - \hat{p}_{t-1}\|_1 + \frac{b_{rot}}{N}\sum_{t=1}^{N}\|\hat{r}_t - \hat{r}_{t-1}\|_1 \tag{9}$$

where p_t, r_t are the ground truth global position and global rotation at step t, while $\hat{p}_t, \hat{r}_t$ are the predicted output. N denotes the length of the predicted sequence. $a_{pos}, a_{rot}, b_{pos}$ and b_{rot} are the coefficients for the loss functions.

3.3.1 Control Conditions

To enhance the staging transformer with shape and target pose information, we initially consider concatenating the tooth's shape code and target pose with its source pose directly (Fig. 3a). However, this approach introduces significant data redundancy since both the shape code and target pose remain unchanged throughout the staging process. Moreover, the disparity in dimensionality risks overshadowing crucial pose details. To resolve these issues, we propose integrating the shape codes $\{c_i \in \mathbb{R}^{108}\}_{i=0}^{27}$ and the target poses $\{x_{tgt}^i \in \mathbb{R}^9\}_{i=0}^{27}$ of 28 teeth into the staging transformer as control conditions, following principles from [43]. Specifically, for each tooth i, we segment its shape code c_i into twelve 9-dimensional vectors $\{c_{i,j} \in \mathbb{R}^9\}_{j=0}^{11}$. These segments, along with the tooth's target pose x_{tgt}^i, are prepended to its pose sequence $\{x_n^i \in \mathbb{R}^9\}_{n=0}^{N}$ along the temporal dimension, where N indicates the number of staging steps to be estimated. This process creates a new input sequence in $\mathbb{R}^{(14+N)\times 9}$ (see Fig. 3b). To facilitate subsequent attention computation, we append a binary mask $m \in \mathbb{Z}^{(14+N)\times 1}$ to the input sequence. This mask indicates the availability of pose information, with 1 denoting known and 0 denoting unknown pose information.

3.3.2 Spatial and Temporal Self-Attention

Our transformer's effectiveness stems from the spatial-temporal attention mechanism. Both types of attention conform to the standard attention framework, as formulated below:

$$MHSA_*(x) = \text{Concat}(\text{head}_*^1, ..., \text{head}_*^h)W^O \tag{10}$$

$$\text{head}_*^i = \text{Attention}_*(xW_i^Q, xW_i^K, xW_i^V) \tag{11}$$

$$\text{Attention}_*(Q,K,V) = \text{softmax}(QK^T + B_*)V \tag{12}$$

where B_* denotes the relative positional bias matrix related to the number of heads h and the type of attention. For spatial attention $MHSA_{space}$, $B_{space} \in \mathbb{R}^{28\times 28}$ represents the Spatial Layout Positional Encoding, derived from the adjacency

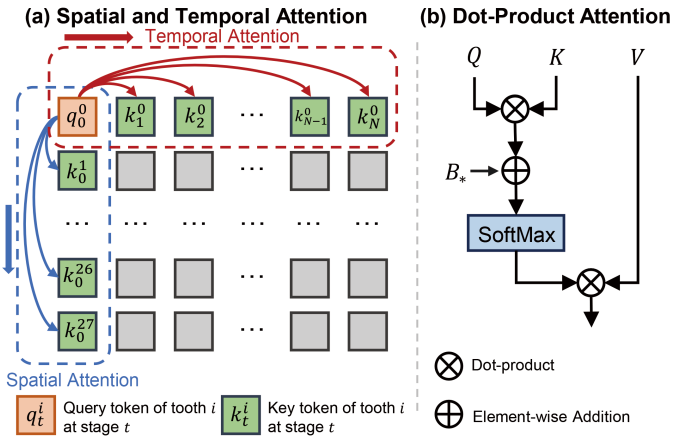


Fig. 4. Illustration of the spatial and temporal attention.

between 28 teeth. And Temporal Relative Positional Encoding $B_{time} \in \mathbb{R}^{(14+N)\times(14+N)}$ for temporal attention $MHSA_{time}$ encodes the relative distance between any two time steps.

As mentioned in Sec 3.2, our shape code, obtained from independently encoded teeth, lacks modeling of inter-tooth dependencies. We bridge this gap with spatial attention, which is well-suited for describing spatial relationships. Our spatial attention focuses solely on the relative spatial positions of different teeth within the same time step. To further enhance the spatial dependencies among teeth, we introduce a spatial layout encoding B_{space} based on the prior knowledge of intrinsic teeth arrangement on the dental arch. We fuse this spatial layout positional encoding into attention score through element-wise addition (detailed in Sec 3.3.3).

Orthodontic staging predicts the dental states at different time steps, which is essentially a sequence prediction task. Hence, we naturally introduce temporal attention to capture correlations between these steps. Notably, to preserve the relative spatial relationships of input and output in each transformer layer, our temporal attention blocks are designed to process only the sequential states of a single tooth, rather than aggregating the states of all teeth across varying time steps. The output of the spatial attention block serves as the input for the temporal attention block. Similar to spatial layout positional encoding, we employ an additional temporal relative positional encoding strategy B_{time} to effectively represent the sequential order of time steps and enable a more nuanced understanding of the temporal dynamics (see details in Sec 3.3.3).

3.3.3 Positional Encoding

To compensate for the transformer's inherent lack of sequential order, we adopt additional positional encodings to extract the spatial and temporal information of teeth. Inspired by [44], we introduce a non-learned relative positional encoding strategy. It involves calculating relative positions between any two tokens and multiplying them by a head-specific scalar μ to form a positional bias matrix. The matrix is then added element-wise to the attention scores, bypassing dimensional normalization. We design two distinct relative positional correlations (i.e., $\rho_{space}, \rho_{time}$) for spatial and temporal attention, respectively, and then encode them to obtain spatial layout encoding B_{space} and temporal relative encoding B_{time}.

Intuitively, the movement of each tooth is influenced by its neighboring teeth in the spatial domain, with closer teeth having

a greater impact. To address this, we propose a correlation term $\rho_{space}(i,j)$ which quantifies the number of teeth between tooth i and tooth j. The term is multiplied by -1 to signify a stronger correlation when there are fewer intervening teeth. We assign numbers to the teeth from 0 to 27, with 0-13 representing upper jaw teeth and 14-27 representing lower jaw teeth. Additionally, the correlation is encoded by a positive head-specific scalar μ, allowing for the incorporation of distinct positional information in different heads.

$$B_{space}(i,j) = \mu \cdot \rho_{space}(i,j) \tag{13}$$

$$\rho_{space}(i,j) = -d_{ij} \tag{14}$$

where $B_{space}(i,j)$ denotes the spatial positional encoding derived from the correlation $\rho_{space}(i,j)$ between tooth i and tooth j. The variable d_{ij} indicates the number of teeth between tooth i and tooth j.

Temporally, our orthodontic staging sequences progress from misaligned to aligned states, so that the sequential order over the temporal dimension is crucial. Each treatment step is intrinsically linked to others. It is observable that one step is more similar to its adjacent steps and less to distant ones. We represent the temporal positional relationships by calculating relative distances $\rho_{time}(i,j)$ between two steps i and j. Similar to spatial positional encoding, we multiply temporal relative distances $\rho_{time}(i,j)$ by a scalar μ to get temporal positional encoding B_{time}. Unlike Alibi encoding, which is unidirectional and masks future steps to focus on past ones, our staging model equally considers both preceding and subsequent states. We hence eschew attention masks and calculate distances without absolute functions, allowing for distinguishing forward and backward positions in the sequence. Our experiments demonstrate that this directed positional encoding method yields superior performance.

Notably, our sequence begins with a fixed set of control conditions comprising 12 shape encoding tokens and a target state. Directly calculating the relative positions diminishes the influence of the control conditions on subsequent tokens. To address this, we set the relative distance between each step and the control conditions to zero, ensuring that the positional matrix exclusively covers the dental state sequence. The temporal positional encoding $B_{time}(i,j)$ between step i and step j can be obtained as follows:

$$B_{time}(i,j) = \begin{cases} 0 & \text{if } i < 13 \text{ or } j < 13 \\ \mu \cdot \rho_{time}(i,j) & \text{otherwise} \end{cases} \tag{15}$$

$$\rho_{time}(i,j) = i - j \tag{16}$$

4 EXPERIMENTS

4.1 Implementation Details

4.1.1 Network Details

For the tooth shape encoder $\mathbf{G}$, we adapt the PoinTr encoder to output 108-dimensional features, where its Geometry-aware Transformer Encoder has 6 heads and a depth of 6 layers. For the staging generation network S, the encoder S_E maps the 10-dimensional input into a 56-dimensional space with a 56-dimensional hidden layer. The transformer layers S_{trans} for the generation have 8 heads and 6 layers in depth. The decoder S_D is similar to the encoder, mapping the 56-dimensional data into a 9-dimensional space as the output.

4.1.2 Training Details

The shape encoder $\mathbf{G}$ is trained from scratch on a subset of our dental model dataset with a batch size of 32 for 200 epochs. The dataset consists of 42,868 samples. We then freeze its parameters and train the Staging Transformer S for maximum 100 epochs with a mini-batch of 16 clips. Each clip is retrieved by sliding a window of 31 from the datasets. In each batch, we randomly sample the prediction length from 5 to 30. We use the Adam optimizer and Noam learning rate scheduler to train our model on one RTX 3090 GPU. The initial learning rate is set to 0.05 with an 8000-iteration warm-up. Throughout our experiments, the loss weights are a_{pos} = 0.1, a_{rot} = 1.0, b_{pos} = 0.1 and b_{rot} = 1.0.

4.1.3 Datasets

We acquire data from 10,000 real-world orthodontic cases provided by an aligner company. Each case includes a 3D dental model of the initial misaligned dentition and a sequence of orthodontic poses that progressively move the teeth towards alignment. It is important to note that the data excludes any tooth shape alterations during orthodontic treatment stages, such as those resulting from interproximal reduction (IPR). This differs slightly from reality, as modifying tooth shape is sometimes necessary to prevent inter-tooth collisions during treatment. We split the dataset into training, validation and test sets by 8:1:1. The number of standard teeth is 28, and there are 2693 cases with missing models in the dataset. Sampling on the training set yields 14,7394 clips, derived from a sliding window with a window size of 31 steps and an offset of 1 step. The validation set contains 6142 clips, obtained with a window of 31, offset by 5 steps. We don't sample on the test set for long-term prediction.

4.2 Evaluation Metrics

We follow L2P and L2Q [45] to evaluate the performance of our network predictions. L2P and L2Q denote the average L2 distance of global position and global quaternion rotation per step. We normalize the data before calculating L2P and L2Q. In addition, we define Mean Position Error (MPE) and Mean Rotation Error (MRE) to compute each tooth's position error and angle error at each step without normalization.

In our study, we evaluate both short-term and long-term prediction utilizing the metrics above. We assess short-term forecast accuracy for prediction lengths from 5 to 30 on the validation set. For a comprehensive evaluation of the entire sequence, we conduct iterative predictions on the test set with an offset of 25 steps and evaluate the metrics based on the length of predicted sequences. Apart from the aforementioned metrics, we calculate the difference ΔN between the lengths of the predicted sequences and the corresponding clinical staging pathways.

4.3 Results and Comparisons

4.3.1 Qualitative Results

Our model is designed to output short-term pathways, while the actual staging process often corresponds to longer steps. We hence provide a complete staging sequence in an iterative manner. The process initiates from a source state x_{src} with the model forecasting a series of steps per iteration. The final step of each iteration is then set as the initial state for the subsequent iteration. This cycle continues until the predicted outcome closely approximates the target state, adhering to predefined thresholds (a maximum

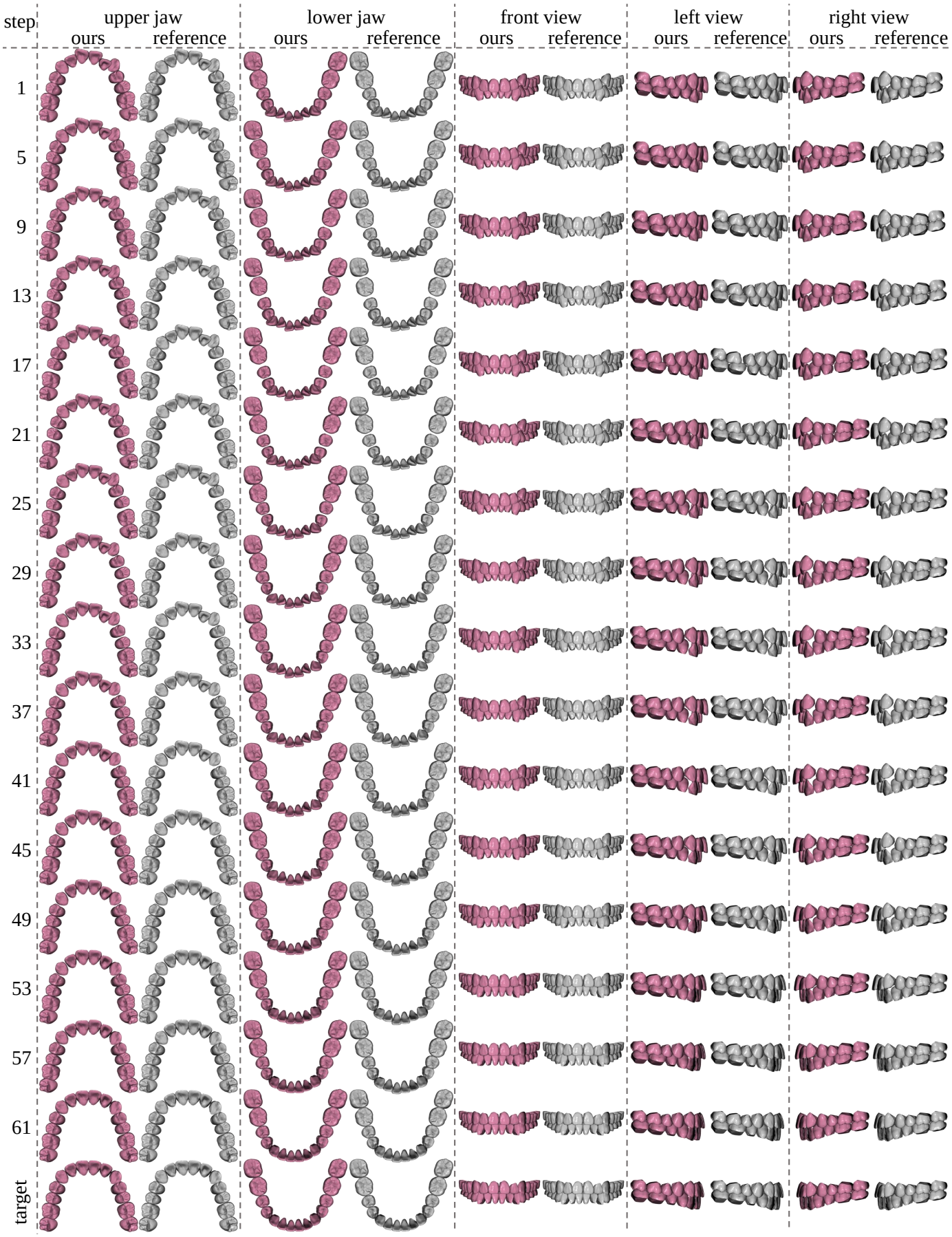


Fig. 5. Visual effects of iterative full-sequence prediction. Our predicted outcomes achieve the target state at step 65 (rendered in pink), while the clinical reference pathway reaches it at step 64 (in gray). Both staging pathways prioritize repositioning the molars and then premolars to secure space for adjusting the misaligned anterior teeth.

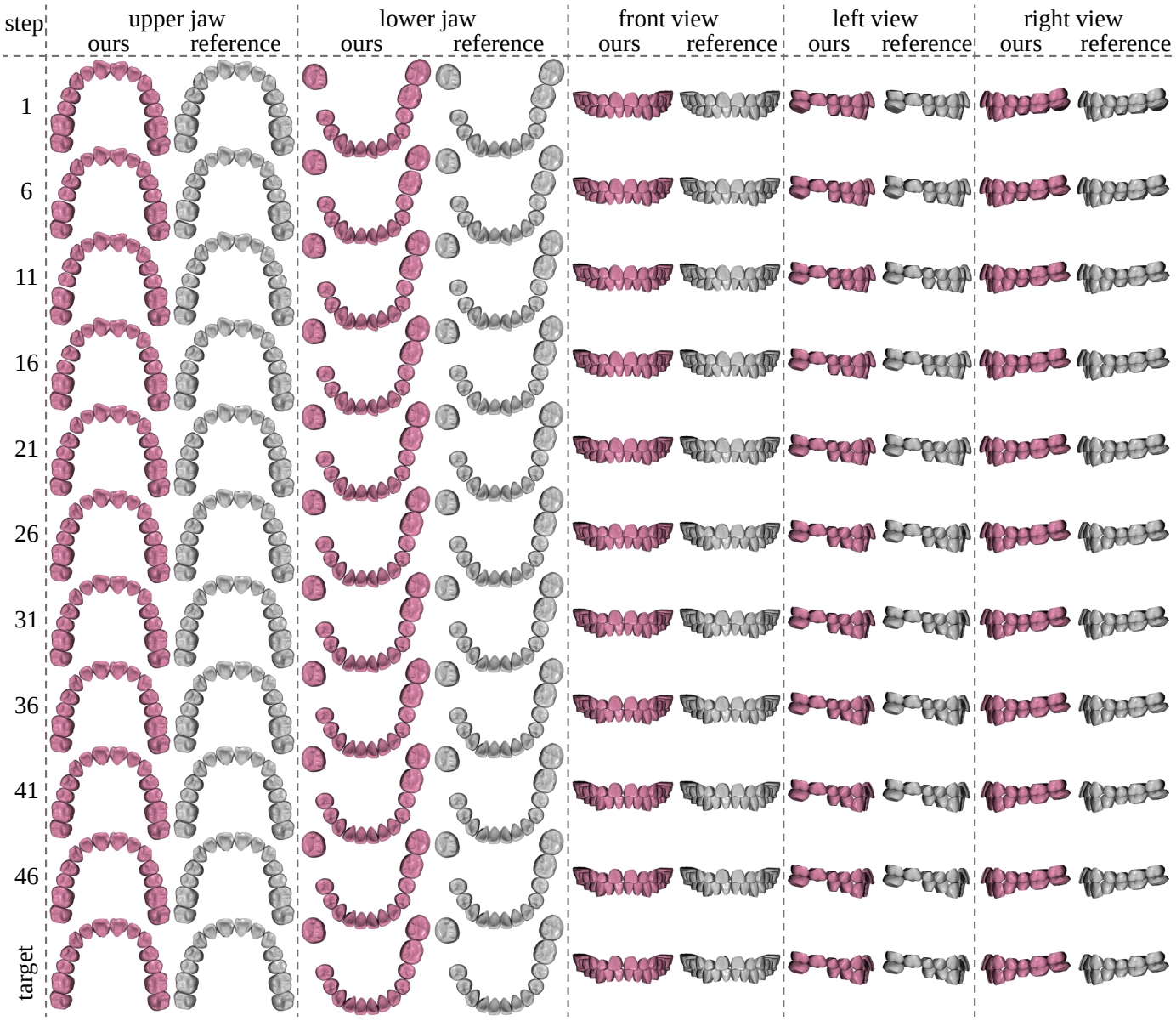


Fig. 6. Visual effects of iterative full-sequence prediction with a missing first molar on the lower left side. Our predicted outcomes achieve the target state at step 47 (rendered in pink), while the clinical reference pathway reaches it at step 51 (in gray). Both pathways concentrate on shifting the molars first and then the premolars to facilitate the adjustment of anterior teeth.

displacement of 0.5mm per tooth per step and a maximum rotation of 3° per step). Fig. 5, Fig. 6 and Fig. 7 illustrate our iterative prediction outcomes and clinical reference pathways. In these cases, we show key steps in the sequence, displaying the top, front, and side views of the upper and lower teeth. We observe that in cases of severe crowding and misalignment, the trend of the predicted movements aligns closely with the reference paths, with a comparable number of staging steps. For more visual effects, please refer to the videos in the supplementary materials.

4.3.2 Comparisons

To demonstrate the superiority of our Transformer-based method, we compare it with the interpolation baseline (Interp) [41], the recent optimization-based improved Gray Wolf Optimization method (IGWO) [6] and the concurrent tooth motion diffusion model (TMDM) [7] in terms of quality and quantity. The interpolation method performs linear interpolation for tooth positions and Spherical Linear Interpolation (SLERP) for tooth rotations. The IGWO method minimizes the total sum of tooth displacement distances and rotation angles while utilizing oriented bounding boxes for tooth collision detection. The TMDM method leverages a diffusion process for prediction and integrates tooth latent representation with graph-based multi-tooth collaboration. These methods necessitate manually setting the number of staging steps, aligning with the reference pathways for consistency. As shown in Table 1, our method demonstrates a reduced deviation from the reference pathways compared to the first two competing methods which are not data-driven, thereby indicating better real-world applicability. Additionally, our method outperforms the diffusion-based approach on all evaluated metrics.

To validate our iterative strategy, we conduct comparisons of direct regression based on our Staging Transformer (TransReg). We train and test the model with a sliding window of length 128, to cover various orthodontic sequences. Short sequences are padded

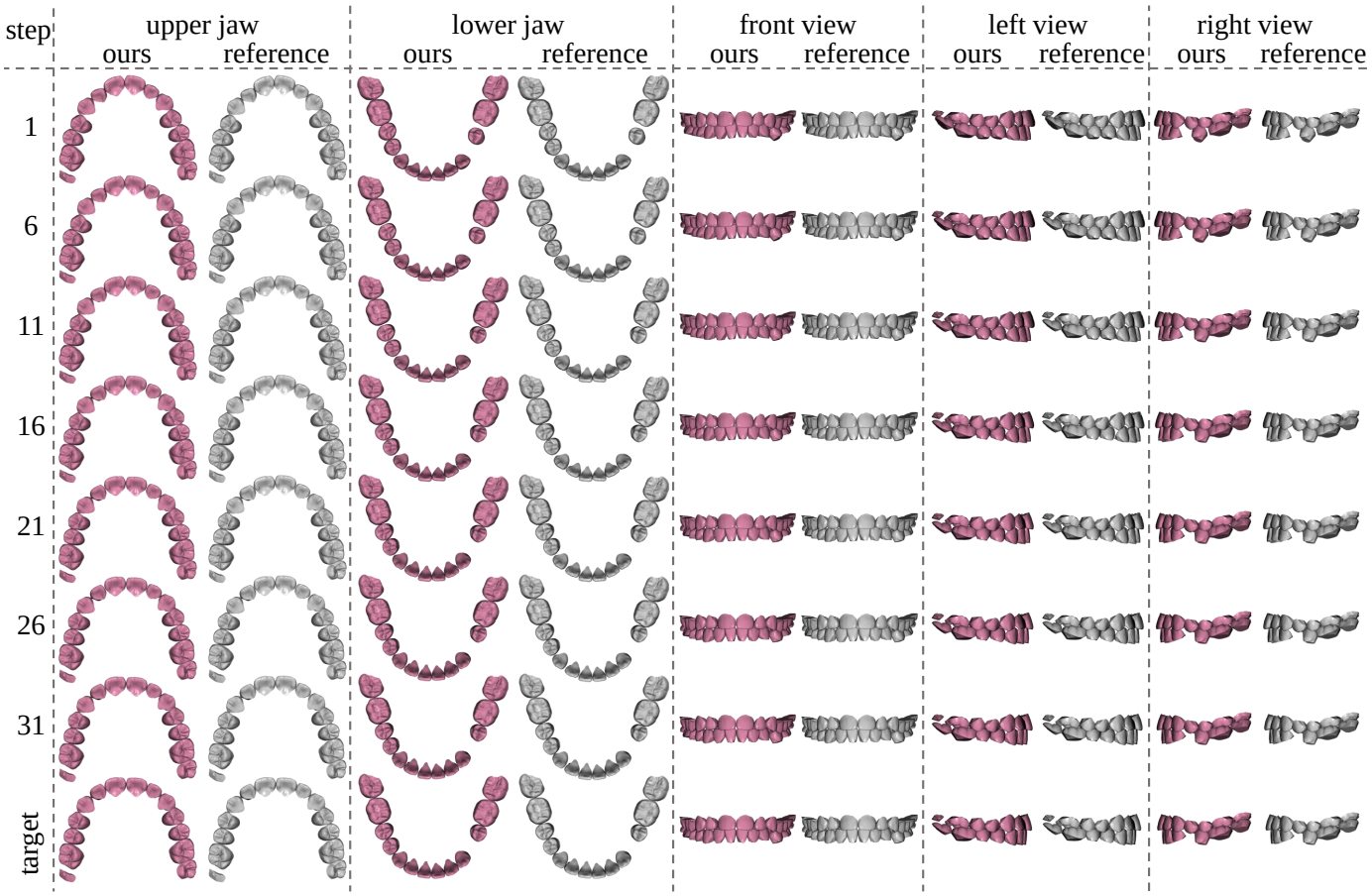


Fig. 7. Visual effects of iterative full-sequence prediction with a missing first premolar on the lower right side. Our predicted outcomes achieve the target state at step 34 (rendered in pink), while the clinical reference pathway reaches it at step 36 (in gray). Both paths involve sequentially moving the molars, premolars, canines, and incisors.

TABLE 1
Comparison with other methods

Method	L2P		L2Q		MPE(mm)		MRE(°)		ΔN		Time Cost(s)	
Interp [41]	1.40	(+42.86%)	0.19	(+26.67%)	0.51	(+82.14%)	3.15	(+61.54%)	\		**0.101**	(-55.11%)
IGWO [6]	1.42	(+44.90%)	0.20	(+33.33%)	0.50	(+78.57%)	3.48	(+78.46%)	\		545.3	(+242255%)
TMDM	1.06	(+8.16%)	0.19	(+26.67%)	0.36	(+28.57%)	2.99	(+53.33%)	\		21.28	(+9358%)
TransReg	1.01	(+3.06%)	**0.15**	(+0.00%)	0.29	(+3.57%)	2.02	(+3.59%)	**9.19**	(-23.35%)	0.124	(-44.89%)
Ours	**0.98**		**0.15**		**0.28**		**1.95**		11.99		0.225	

TABLE 2
Comparison of Collision Frequency

	Threshold	Interp	IGWO	TMDM	TransReg	Ours	Reference
Overall Collision Frequency	0.5mm	0.0415	0.0408	0.0402	0.0406	0.0397	0.0382
	0.3mm	0.1149	0.1088	0.0994	0.1014	0.0996	0.0895
	0.1mm	0.2322	0.2286	0.1886	0.1851	0.1830	0.1436
Tooth#1 Collision Frequency	0.1mm	0.3095	0.2731	0.0557	0.0166	0.0035	0.0000
Tooth#2 Collision Frequency	0.1mm	0.1814	0.0934	0.0329	0.0310	0.0126	0.0000
Tooth#3 Collision Frequency	0.1mm	0.0944	0.0731	0.0075	0.0146	0.0064	0.0000

with the last step. As indicated in the fourth and the fifth rows of Table 1, although the iterative approach marginally increases both time cost and length error, it enhances prediction accuracy.

Figure 8 offers a visual comparison of a case of lower tooth crowding, showcasing top and front views. The interpolation method ignores the spatial inter-tooth correlation, proceeding with tooth movement even in constrained spaces. The TMDM method shows some capability in learning movement patterns that initially shift certain teeth to create sufficient space. However, since its diffusion model relies on GRUs, TMDM has limitations in processing longer sequences, which hinders its ability to precisely coordinate the planning of tooth movements. And the pathways produced by these two methods both result in tooth intersections. Our method, in contrast, learns teeth movement patterns from extensive training cases and adeptly manages the order of tooth movements, ensuring adequate space for dental crowding. Specifically, the iterative short-term prediction framework benefits from accommodating multiple solutions during training and testing. This capability provides a clear advantage in managing orders of tooth movements over frameworks that directly predict long sequences.

Collision Avoidance Avoiding interdental collisions in the planning of tooth movement paths is crucial. To evaluate the

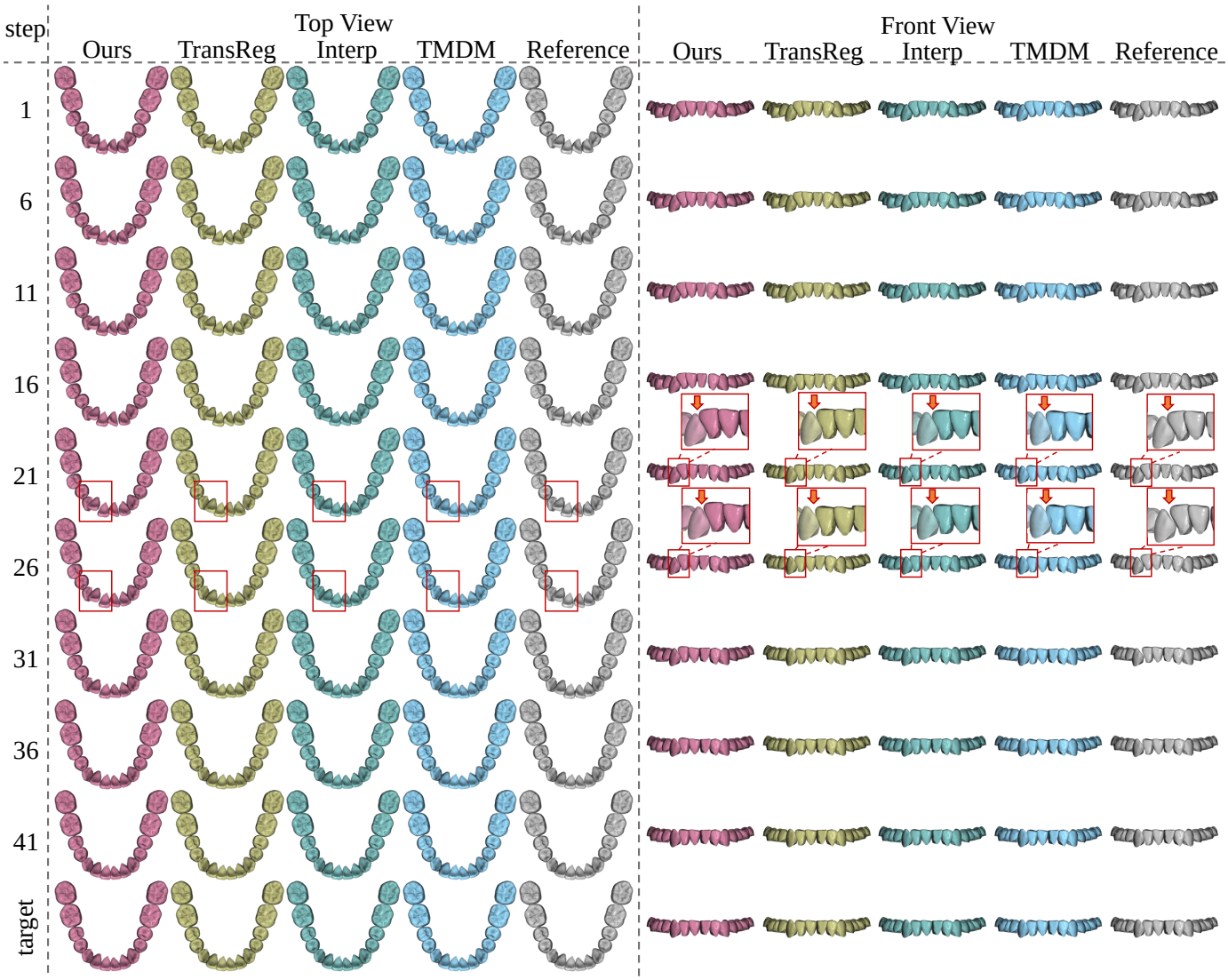


Fig. 8. A visual comparison of several methods. Our method iteratively predicts a path length of 43 steps, while the direct regression method using a Transformer (TransReg) produces 39 steps. Both interpolation (Interp [41]) and diffusion-based model (TMDM [7]) manually set the path length to 44 steps, matching the clinical reference. We present only the top and front views of the lower teeth. Until step 21, our method and the reference path shift the right-side teeth to accommodate the crowded central incisors, lateral incisors, and canines. TransReg rapidly repositions the right-side teeth as well; however, the left lateral incisor and the left canine move before the right-side teeth completely vacate the area. In contrast, Interp opts for the shortest path, moving teeth incrementally in crowded situations instead of creating space. TMDM moves the right-side teeth almost simultaneously, rather than in sequential batches. As a result, the left incisor begins to move earlier compared to other methods. At steps 21 and 26, the left lateral incisor and left canine models in the TransReg, Interp and TMDM methods appear to intersect, unlike our method and the reference path, which maintain spacing (highlighted in red box for emphasis, with the left canine rendered semi-transparently).

performance of various methods in collision avoidance, we compute the average overall collision frequency, defined as collision count$/$(step count$\times$tooth count). Specifically, we extract the convex hull of each tooth and assess collision using the Gilbert-Johnson-Keerthi (GJK) algorithm from the hpp-fcl library [46], with penetration depth thresholds of 0.1mm, 0.3mm and 0.5mm. The results are exhibited in Table 2. It can be seen that collisions are detected even in the reference paths. This is due to the simplification of tooth shape consistency, as explained in Section 4.1.3 - Datasets. To enhance clarity, we specifically select some teeth that are collision-free along the reference pathway to evaluate the robustness of the proposed algorithm in collision avoidance (as shown in the last three rows of Table 2). As indicated in Table 2, despite the lack of a dedicated mechanism for collision avoidance, our method demonstrates robustness against collisions throughout the staging process. It learns tooth movement patterns from reference paths, effectively mimicking reference to prevent collisions and yielding results comparable to those paths.

Discontinuity Prevention Orthodontic treatment should avoid abrupt changes in tooth movements. To assess the effectiveness of various methods in preventing such discontinuities, we compute the discontinuity frequency, defined as sudden changes count$/$(step count $\times$ tooth count). A sudden change is defined as a displacement greater than 0.5 mm or a rotation exceeding 3 degrees in a single step. As shown in Table 3, the interpolation method and the reference path exhibit very smooth results, with no detected discontinuities. Both TransReg and our approach, optimized with a smooth loss function, demonstrate superior performance in discontinuity prevention.

TABLE 3
Comparison of Discontinuity Frequency

	Interp	IGWO	TMDM	TransReg	Ours	Reference
Overall Discontinuity Frequency	**0.00000**	0.00351	0.00407	0.00116	0.00023	**0.00000**

4.4 Ablation Study

In this section, we ablate key components of our proposed network model, on the validation set. We assess the performance of different network configurations across prediction lengths from 5 to 30, utilizing metrics such as L2P, L2Q, MPE, and MRE.

4.4.1 Attention Mechanism

To verify the effectiveness of the spatial attention and the temporal attention mechanism, we conduct an ablation study by removing each attention module from each transformer layer.

The comparison between Row 1 and Row 3 in Table 4 reveals that, in the absence of the temporal attention module, removing the spatial attention module yields better performance than including it for shorter sequences (lengths of 5 and 10). However, for longer sequences, the inclusion of spatial attention enhances the network's predictive ability to some extent. With the temporal attention module, the Row 2 vs. Row 4 comparison indicates that adding spatial attention consistently improves performance across various sequence lengths, significantly reducing the metrics. These results highlight the critical role of inter-tooth spatial interactions and the foundational role of inter-step temporal relationships in the predictive accuracy of the network.

When assessing the temporal attention module (Row 1 vs. Row 2, Row 3 vs. Row 4), the staging transformer equipped with temporal attention greatly surpasses its counterpart without the module. The temporal attention module achieves a more significant reduction in performance metrics compared to the spatial attention module. These findings underscore the importance of the temporal attention module for handling time series data.

Particularly, when spatial attention is already included (Row 3 vs. Row 4), the temporal attention module provides a greater boost to prediction performance than in spatial attention's absence (Row 1 vs. Row 2). This suggests that the synergy between the two modules leads to enhanced predictive capabilities of our Transformer.

4.4.2 Tooth Shape Encoding

To prove the indispensability of shape codes, we exclude them from the staging transformer's input. We also compare the performance of shape codes generated by various tooth shape encoders. Specifically, FoldingNet [14] employs a graph-based encoder, whereas SnowflakeNet [23] and our utilized PoinTr [18] both incorporate transformer-based encoders. As indicated in Table 5, shape codes produced by PoinTr slightly enhance the predictive performance of the network. Contrarily, SnowflakeNet and FoldingNet fail to exhibit any superior performance over configurations devoid of shape codes across various metrics, contracting our initial hypothesis. We think that the spatial attention mechanism, devised to offset the limitations of shape codes in modeling tooth interaction dynamics, assumes a primary role in prediction, relegating shape codes to a secondary position. Consequently, we conduct another ablation study for shape codes on networks without the spatial attention module. Table 6 reveals that, in the absence of spatial attention, employing shape codes remarkably excels over not using them. Among the encoding methods, shape codes generated by PoinTr demonstrate superior overall efficacy compared to the other two methods. The findings presented in Table 5 validate our aforementioned hypothesis.

4.4.3 Positional Encoding

We investigate the effectiveness of our proposed spatial layout positional encoding and temporal relative positional encoding by removing or replacing the Staging Transformer's positional encodings. The last row in Table 7 shows the complete model. The blank entries in the first three rows indicate the absence of specific positional encodings. Compared to the baseline without any positional encoding (first row), incorporating either spatial (second row) or temporal (third row) positional encoding enhances performance, particularly noting that temporal encoding exerts a more pronounced effect. Spatial encodings are less impactful for longer prediction tasks. The integration of both positional encodings (last row) significantly surpasses using either encoding alone (second row and third row). On top of spatial encoding cooperating with temporal encoding offers more noticeable performance benefits than without temporal encoding's help. The 'sin' notation in the fourth row represents two-dimensional sinusoidal absolute positional encoding, leading to reduced efficacy. The results reveal the superiority of relative positional relationships over absolute ones across spatial and temporal dimensions. In the fifth row, while maintaining our spatial positional encoding, we adopt an ALiBi [44] variant as temporal positional encoding which calculates absolute values for temporal relative distances without the standard ALiBi's attention mask. Our approach to temporal positional encoding, differentiating past from future steps, demonstrates a slight improvement over the undirected ALiBi variant.

4.4.4 Control Conditions

Our control condition module involves three key elements: shape codes, target state, and the fusion strategy. In Sec 4.4.2, the effectiveness of shape codes has been verified. Herein, we evaluate the critical role of the target state and the effectiveness of our fusion approach. we first remove the target state from the input, thereby limiting the network's awareness to only the tooth source state and omitting the desired aligned poses. The second row in Table 8 shows that the lack of the target state's guidance significantly deteriorates the predictive accuracy of the network. Conversely, incorporating the target state narrows down the solution space, resulting in more plausible and precise outcomes. To fuse into our Staging Transformer, we do not merely concatenate the control conditions to each tooth's pose vector at each time step. Instead, we transform and place them at the forefront of the sequence. This strategy enables the control conditions to interact with other tokens and reduces the redundant features. The third row corroborates our fusion strategy's advantage over direct concatenation.

4.5 User Study

We organize a user study to evaluate the quality of our results. We randomly selected 18 orthodontic staging sequences from the test set and generated a group of predictions by our networks. These random cases with lengths ranging from 28 to 80 include scenarios with tooth extraction and model missing. We randomized the order of presentation for each case's prediction and clinical reference

TABLE 4
Ablation of Attention Mechanism

Attention Modules		**L2P**				**L2Q**				**MPE(mm)**				**MRE(°)**			
$MHSA_{space}$	$MHSA_{time}$	5	10	20	30	5	10	20	30	5	10	20	30	5	10	20	30
		0.74	1.07	1.72	2.29	0.10	0.15	0.24	0.31	0.22	0.39	0.52	0.73	1.41	2.17	3.59	4.80
	✓	0.64	0.85	1.23	1.41	0.09	0.13	0.18	0.19	0.20	0.25	0.37	0.44	1.26	1.82	2.59	2.74
✓		0.78	1.00	1.55	2.08	0.13	0.16	0.23	0.29	0.22	0.28	0.45	0.64	1.72	2.26	3.43	4.53
✓	✓	**0.42**	**0.62**	**0.88**	**1.00**	**0.07**	**0.10**	**0.14**	**0.15**	**0.11**	**0.17**	**0.24**	**0.28**	**0.96**	**1.42**	**1.89**	**1.96**

TABLE 5
Ablation of Tooth Shape Encoding

		L2P				**L2Q**				**MPE(mm)**				**MRE(°)**			
		5	10	20	30	5	10	20	30	5	10	20	30	5	10	20	30
Tooth Shape Encoder	PoinTr	**0.42**	**0.62**	**0.88**	**1.00**	**0.07**	**0.10**	**0.14**	**0.15**	**0.11**	**0.17**	**0.24**	**0.28**	**0.96**	**1.42**	**1.89**	**1.96**
	SnowflakeNet	0.44	0.64	0.89	1.01	**0.07**	0.11	**0.14**	**0.15**	0.13	**0.17**	0.25	0.29	1.01	1.45	1.93	2.01
	FoldingNet	0.45	0.64	0.89	**1.00**	**0.07**	0.11	**0.14**	**0.15**	0.12	**0.17**	0.25	**0.28**	**0.96**	1.44	1.93	2.00
w/o tooth shape encoder		0.43	0.63	0.89	1.01	**0.07**	0.11	0.15	0.16	0.12	**0.17**	0.25	0.29	0.97	1.45	1.95	2.03

TABLE 6
Ablation of Tooth Shape Encoding **without Spatial Attention**

		L2P				**L2Q**				**MPE(mm)**				**MRE(°)**			
		5	10	20	30	5	10	20	30	5	10	20	30	5	10	20	30
Tooth Shape Encoder	PoinTr	0.64	**0.85**	**1.23**	**1.41**	0.09	**0.13**	**0.18**	**0.19**	0.20	**0.25**	**0.37**	**0.44**	1.26	**1.82**	**2.59**	**2.74**
	SnowflakeNet	**0.61**	0.86	1.28	1.43	**0.08**	**0.13**	**0.18**	0.20	**0.19**	0.26	0.39	0.45	**1.22**	1.86	2.67	2.85
	FoldingNet	0.71	**0.85**	1.25	1.43	0.09	**0.13**	**0.18**	0.20	0.22	**0.25**	0.38	0.45	1.25	**1.82**	2.62	2.76
w/o tooth shape encoder		0.81	0.97	1.31	1.51	0.10	0.15	0.19	0.21	0.28	0.32	0.44	0.52	1.49	2.02	2.79	2.95

TABLE 7
Ablation of Positional Encoding

Positional Encodings		**L2P**				**L2Q**				**MPE(mm)**				**MRE(°)**			
B_{space}	B_{time}	5	10	20	30	5	10	20	30	5	10	20	30	5	10	20	30
		0.78	0.88	1.34	1.69	0.09	0.14	0.20	0.24	0.22	0.24	0.38	0.50	1.24	1.82	2.82	3.37
✓		0.64	0.85	1.32	1.66	0.09	0.13	0.20	0.23	0.20	0.23	0.38	0.49	1.18	1.80	2.76	3.26
	✓	0.54	0.69	0.96	1.08	0.08	0.11	0.15	0.16	0.15	0.18	0.26	0.30	1.02	1.51	2.04	2.11
sin	sin	1.08	0.67	0.89	1.01	0.10	0.11	**0.14**	**0.15**	0.40	0.21	0.27	0.31	1.58	1.50	1.97	2.05
✓	ALiBi variant [44]	0.44	0.64	0.91	1.02	**0.07**	0.11	**0.14**	**0.15**	0.12	**0.17**	0.25	0.29	**0.96**	1.43	1.93	2.00
✓	✓	**0.42**	**0.62**	**0.88**	**1.00**	**0.07**	**0.10**	**0.14**	**0.15**	**0.11**	**0.17**	**0.24**	**0.28**	**0.96**	**1.42**	**1.89**	**1.96**

TABLE 8
Ablation of Control Conditions

	L2P				**L2Q**				**MPE(mm)**				**MRE(°)**			
	5	10	20	30	5	10	20	30	5	10	20	30	5	10	20	30
w/ x_{tgt}	**0.42**	**0.62**	**0.88**	**1.00**	**0.07**	**0.10**	**0.14**	**0.15**	**0.11**	**0.17**	**0.24**	**0.28**	**0.96**	**1.42**	**1.89**	**1.96**
w/o x_{tgt}	0.52	0.84	1.34	1.72	0.08	0.13	0.20	0.24	0.15	0.24	0.40	0.55	1.14	1.92	3.16	3.93
direct concatenation	0.48	0.78	1.26	1.60	0.08	0.13	0.19	0.22	0.13	0.21	0.36	0.48	1.08	1.71	2.67	3.20

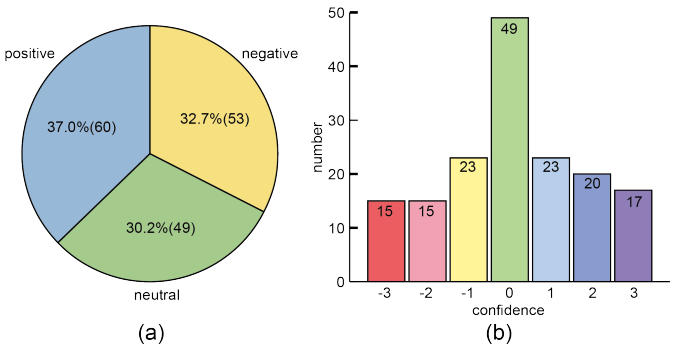


Fig. 9. User Study Results. (a) The user preference for the prediction results versus the clinical reference. (b) The confidence ratings that users give to their selections.

pathway. We invited 9 professional orthodontists to compare and select the better one between the prediction and clinical reference pathways for each case. As shown in Figure 9(a), 37.0% of ratings show that participants favor the results generated by our network, and 30.2% think our results are equal to the reference. To evaluate the gap between the predicted results and reference pathways, we also asked the participants to score the confidence of each selection on a scale of 0 to 3. A score of 3 indicates that the selected one is undoubtedly better than the other, while a score of 0 indicates that it is difficult to judge which is better. Finally, we combine the scores with the selections, setting the signs of scores to negative if the orthodontists prefer the reference pathway. As shown in Figure 9(b), professional orthodontists find it hard to distinguish between our staging results and the clinical reference in many cases. The weighted scores are averaged and normalized to the range of [-1,1], with 1 indicating that our results are better than the clinical reference pathways and 0 indicating equal quality. Our final score is 0.03, which demonstrates the ability of our method to generate reliable results once again.

5 Limitations and Future Work

Here, we address several limitations inherent in our methodology. Firstly, the total length of the staging predicted iteratively may not perfectly align with the clinical reference length. Our iterative approach, aimed at approximating the target state by predicting intermediate steps, concludes predictions based on a manually set threshold. However, this threshold might not universally apply, leading to discrepancies in cases where the reference paths involve large step-wise movements and cease before meeting the threshold. Consequently, our results may involve more steps in such scenarios. Additionally, our method typically moves teeth shorter distances directly towards the target, minimizing excessive back-and-forth movements, in contrast to the reference paths observed in complex scenarios requiring re-adjustment. Notably, in our user study, orthodontists expressed a preference for the results provided by our approach in such cases. Secondly, we have observed instances of teleportation and stalling artifacts, albeit in a minority of cases. To enable parallel computation during training, we extend shorter sequences by repeating the last step. This approach may inadvertently lead the network to prematurely halt new movements if it perceives the current state as close to the target but not meeting the termination criteria. The absence of precise time-to-arrival further compounds the emergence of these artifacts. Thirdly, while our model learns from reference paths and demonstrates robustness against collisions, it falls short of achieving collision-free outcomes. This limitation arises from the exclusion of physical collision losses, which penalize tooth-to-tooth contacts, from the training process due to their high computational demand. Future endeavors will explore the integration of explicit collision constraints to enhance avoidance capabilities. Lastly, orthodontic staging necessitates a comprehensive consideration of various factors such as tooth morphology, the relationship between tooth roots and the alveolar bone, and oral health. However, our current method overlooks tooth roots by relying solely on crown morphology derived from oral scans, potentially jeopardizing tooth stability. Future research endeavors will focus on integrating Cone Beam Computed Tomography (CBCT) assessments to achieve a more comprehensive understanding of tooth root conditions.

6 Conclusion

In this paper, we propose the first Transformer-based method for orthodontic staging in an iterative manner. By mirroring real-life treatment scenarios through short-term forecasting, our method allows for dynamic adjustments to orthodontic staging and embraces the complexity and multiple potential outcomes. We employ a Transformer to predict tooth movements in short steps and generate a full staging sequence through an iterative process. Our Transformer integrates both spatial and temporal attention mechanisms enhanced with relative positional encodings to accurately model the intricate dynamics of tooth interactions. To enable a more effective utilization of the tooth's morphological features, we extract shape codes from the raw 3D teeth point clouds and inject them into the transformer. Extensive experiments and a user study demonstrate that our method can generate staging sequences comparable with those planned by orthodontists.

References

[1] Z. Li, K. Li, and B. Li, "Research on path planning for tooth movement based on genetic algorithms," in *2009 International Conference on Artificial Intelligence and Computational Intelligence*, vol. 1. IEEE, 2009, pp. 421–424.

[2] Z. Li and G. Yang, "Research on simulation and optimization method for tooth movement in virtual orthodontics," in *Advances in Computer Science, Environment, Ecoinformatics, and Education: International Conference, CSEE 2011, Wuhan, China, August 21-22, 2011. Proceedings, Part I*. Springer, 2011, pp. 270–275.

[3] Z. Li, T. Liu, H.-A. Li, and Z. Sun, "Orthodontic path planning method based on optimized artificial bee colony algorithm," in *Journal of Physics: Conference Series*, vol. 1544, no. 1. IOP Publishing, 2020, p. 012017.

[4] X. XU, P. QIN, and J. ZENG, "Orthodontic path planning based on improved particle swarm optimization algorithm," *Journal of Computer Applications*, vol. 40, no. 7, p. 1938, 2020.

[5] T. Ma, J. Lyu, Q. Yang, Z. Li, Y. Li, Y. Chen, and X. Ren, "Orthodontic Overcorrection Scheme Generation Based on Improved Multiparticle Swarm Optimization," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–12, 2021, publisher: Hindawi Limited.

[6] X. Du, T. Yu, and K. Chen, "An orthodontic path planning method based on improved gray wolf optimization algorithm," *Soft Computing*, vol. 27, no. 22, pp. 16 589–16 609, Nov 2023. [Online]. Available: https://doi.org/10.1007/s00500-023-08924-0

[7] Y. Fan, G. Wei, C. Wang, S. Zhuang, W. Wang, and Y. Zhou, "Collaborative tooth motion diffusion model in digital orthodontics," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 2, pp. 1679–1687, Mar. 2024. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/27935

[8] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[9] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[10] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution On X-Transformed Points," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[11] G. Te, W. Hu, A. Zheng, and Z. Guo, "Rgcnn: Regularized graph cnn for point cloud segmentation," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 746–754.

[12] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019, pp. 9621–9630.

[13] 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 Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019, publisher: ACM New York, NY, USA.

[14] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 206–215.

[15] Y. Zhao, T. Birdal, H. Deng, and F. Tombari, "3D point capsule networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1009–1018.

[16] N. Srivastava, H. Goh, and R. Salakhutdinov, "Geometric capsule autoencoders for 3d point clouds," *arXiv preprint arXiv:1912.03310*, 2019.

[17] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in *2018 international conference on 3D vision (3DV)*. IEEE, 2018, pp. 728–737.

[18] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "PoinTr: Diverse Point Cloud Completion with Geometry-Aware Transformers," 2021. [Online]. Available: https://arxiv.org/abs/2108.08839

[19] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "Topnet: Structural point cloud decoder," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 383–392.

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

[21] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.

[22] J. Lin, M. Rickert, A. Perzylo, and A. Knoll, "Pctma-net: Point cloud transformer with morphing atlas-based point generation network for dense point cloud completion," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5657–5663.

[23] P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, and Z. Han, "Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5499–5509.

[24] X. Wen, P. Xiang, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "Pmp-net++: Point cloud completion by transformer-enhanced multi-step point moving paths," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 852–867, 2022.

[25] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[26] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[27] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International conference on machine learning*. PMLR, 2017, pp. 1243–1252.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

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

[30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[31] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[32] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[33] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," *arXiv preprint arXiv:1809.04281*, 2018.

[34] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 178–186.

[35] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in neural information processing systems*, vol. 32, 2019.

[36] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[37] D. Weissenborn, O. Täckström, and J. Uszkoreit, "Scaling autoregressive video models," *arXiv preprint arXiv:1906.02634*, 2019.

[38] R. Rakhimov, D. Volkhonskiy, A. Artemov, D. Zorin, and E. Burnaev, "Latent video transformer," *arXiv preprint arXiv:2006.10704*, 2020.

[39] Y. Zeng, J. Fu, and H. Chao, "Learning joint spatial-temporal transformations for video inpainting," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*. Springer, 2020, pp. 528–543.

[40] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li, "Fuseformer: Fusing fine-grained information in transformers for video inpainting," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14 040–14 049.

[41] M. Chapuis, M. Lafourcade, W. Puech, G. Guillerm, and N. Faraj, "Animating and adjusting 3d orthodontic treatment objectives," in *GRAPP 2022-17th International Conference on Computer Graphics Theory and Applications*. SCITEPRESS, 2022, pp. 60–67.

[42] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.

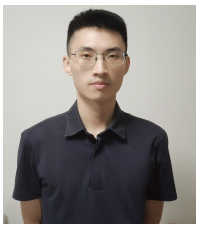
[43] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "Ctrl: A conditional transformer language model for controllable generation," *arXiv preprint arXiv:1909.05858*, 2019.

[44] O. Press, N. A. Smith, and M. Lewis, "Train short, test long: Attention with linear biases enables input length extrapolation," *arXiv preprint arXiv:2108.12409*, 2021.

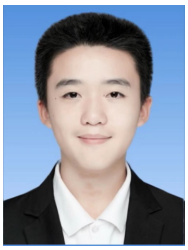
[45] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, "Robust motion in-betweening," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 60–1, 2020, publisher: ACM New York, NY, USA.

[46] J. Pan, S. Chitta, D. Manocha, F. Lamiraux, J. Mirabel, J. Carpentier, L. Montaut *et al.*, "Hpp-fcl: an extension of the flexible collision library," https://github.com/humanoid-path-planner/hpp-fcl, 2015–2023.

Jiayue Ma is a Master candidate at the State Key Lab of CAD&CG, Zhejiang University. She obtained her B.S. degree in Zhejiang University. Her research interests include digital orthodontics and deep learning.



Jianwen Lou is a Researcher at the School of Software Technology, Zhejiang University. He received his Ph.D. degree in Visual Computing from the University of Portsmouth, UK, in 2021. His research interests include 3D geometry modeling and editing.



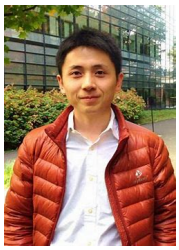
Borong Jiang is a research intern at the State Key Lab of CAD&CG, Zhejiang University. His research interest lies in machine learning and digital geometry.

Hengyi Ye is a research assistant at the State Key Lab of CAD&CG, Zhejiang University. She obtained her B.E. degree from Zhejiang University. Her research interest focuses on Animation Simulation and Computational Geometry.

Wenke Yu is an Associate Chief Physician at the Chengxi Branch of Hangzhou Stomatology Hospital. She received her Master's degree in Orthodontics from Zhejiang University and is a specialist member of the Orthodontics Professional Committee of the Chinese Stomatological Association (COS). Dr. Yu specializes in early preventive orthodontics for children and correction of malocclusions in adolescents and adults. She has trained at the Medical College of Georgia, USA, and is skilled in both traditional fixed and bracketless invisible orthodontic techniques.

Xiang Chen is an Associate Professor in the State Key Lab of CAD&CG, Zhejiang University. He received his Ph.D. in Computer Science from Zhejiang University in 2012. His current research interests mainly include fabrication-aware design, image analysis or editing, shape modeling or retrieval and computer-aided design.

Kun Zhou is a Cheung Kong Professor in the Computer Science Department of Zhejiang University. He received his B.S. degree and Ph.D. degree in computer science from Zhejiang University in 1997 and 2002, respectively. His research interests are in visual computing, parallel computing, human computer interaction, and virtual reality. He currently serves on the editorial advisory boards of ACM Transactions on Graphics and IEEE Spectrum. He is a Fellow of IEEE.



Youyi Zheng is a Researcher at the State Key Lab of CAD&CG, Zhejiang University. He received a BS degree and an MS degree in Mathematics, both from Zhejiang University, China, in 2005 and 2007, and a PhD in Computer Science from the Hong Kong University of Science & Technology in 2011. His research interests include geometric modeling, imaging, and human computer interaction. He has served as an Associate Editor of The Visual Computer and Frontiers of Computer Science.