This MICCAI paper is the Open Access version, provided by the MICCAI Society. It is identical to the accepted version, except for the format and this watermark; the final published version is available on SpringerLink.

Supplementary Material: Teeth2Point

Qi Ma[1,2][0009−0005−4028−6917], Shipra Jain[2][0009−0008−1833−3416], Niko Benjamin Huber[2][0009−0007−2089−0150], and Ender Konukoglu[1][0000−0002−2542−3611]

[1] Computer Vision Lab, ETH Zurich, Switzerland
{fname.lname}@vision.ee.ethz.ch
[2] Align Technology {sjain, nhuber}@aligntech.com

A Implementation Details

Volumetric Baselines. Following the official nnU-Net implementation [3], we used the `nnUNetPlannerResEncM` and `nnUNetPlannerResEncL` plans. The models were trained on NVIDIA A6000 GPUs with a batch size of 2 for 1000 epochs using a polynomial learning-rate schedule with exponent 0.9. We disabled mirroring with `nnUNetTrainerNoMirroring`. For convolutional baselines including nnU-Net and MedNeXt [7], the objective combines Dice loss and cross-entropy loss with equal weights. Auxiliary supervision is applied to decoder outputs at scales 0.5, 0.25, 0.125, and 0.0625. Sliding-window inference uses 25% overlap. Case-level DSC and HD95 are computed per scan and averaged across the test set.

Layer	Module	Input Shape	Output Shape	Parameters × Block
Embed	Conv3D	$(B, 1, D, H, W)$	$(B, 32, D, H, W)$	$K = 3^3$, $S = 1$
Enc.0	Conv3D	$(B, 32, \ldots)$	$(B, 32, \ldots)$	$K = 3^3$, $S = 1$, $\times 1$
Enc.1	Conv3D	$(B, 32, \ldots)$	$(B, 64, D/2, \ldots)$	$K = 3^3$, $S = 1$, $\times 2$
Enc.2	Conv3D	$(B, 64, \ldots)$	$(B, 128, D/4, \ldots)$	$K = 3^3$, $S = 1$, $\times 3$
Enc.3	Conv3D	$(B, 128, \ldots)$	$(B, 256, D/8, \ldots)$	$K = 3^3$, $S = 1$, $\times 5$
Enc.4	Conv3D	$(B, 256, \ldots)$	$(B, 320, D/16, \ldots)$	$K = 3^3$, $S = 1$, $\times 5$
Enc.5	Conv3D	$(B, 320, \ldots)$	$(B, 320, D/32, \ldots)$	$K = 3^3$, $S = 1$, $\times 5$
Dec.	Up-conv + Conv3D	encoder features	segmentation logits	symmetric U-Net decoder

Table S1. Compact architecture summary for the residual nnU-Net baseline. Decoder blocks mirror the encoder and use transpose convolutions for upsampling, following the U-Net design [6].

Point-Based Stage. The point-stage backbone follows the encoder-decoder design of Point Transformer V3 [8]. A linear embedding maps point attributes to tokens. Hierarchical grid pooling increases the grid size across encoder stages and aggregates features by max pooling within each grid cell. During up-projection, low-resolution decoded features are broadcast back to points associated with the corresponding pooled cell and fused with skip features.

The point network is first trained for 400 epochs on ground-truth ROIs, then adapted for 100 epochs on predicted ROIs to improve robustness to first-stage noise. The model uses a batch size of 8, cross-entropy plus Dice loss during

supervised training, and Dice-only fine-tuning in the final noise-aware adaptation stage. Pretraining follows a DINO-style self-distillation objective [1, 2, 5, 9] on 3,224 unlabeled CBCT scans for 200 epochs. The masking ratio follows a cosine schedule from 0.3 to 0.7.

Layer	Module	Channels	Blocks
Embed	Linear	$C_{\text{input}} \to 32$	1
Enc.0	Attention	32	2
Enc.1	Pool + Attention	64	2
Enc.2	Pool + Attention	128	2
Enc.3	Pool + Attention	256	6
Enc.4	Pool + Attention	512	2
Dec.3–Dec.0	Up-project + Skip + Attention	256, 128, 64, 64	2 each
Head	Linear	$64 \to C_{\text{class}}$	1

Table S2. Compact VAP backbone configuration used by Teeth2Point. For a representative tooth ROI, the point counts across encoder stages are approximately $[113541, 17970, 3254, 671, 160]$.

Augmentation	Domain	Description	Probability
Random dropout	CBCT/points	Drop 5% of foreground points	0.1
Random rotation	CBCT/points	Rotate around each axis within $[-1.9°, 1.9°]$	0.2
Random scaling	Points	Scale coordinates by $[0.8, 1.2]$	1.0
Random flip	CBCT/points	Flip left-right tooth positions and labels	0.2
Elastic distortion	CBCT/points	Smooth random non-rigid displacement field	0.9
Gamma transform	CBCT	Nonlinear intensity transform with $\gamma \in [0.8, 1.2]$	0.1
Metal-artifact intensity	CBCT	Intensity amplification in selected tooth regions	0.1

Table S3. Training augmentations used for the point-based VAP stage.

B Additional Comparisons

ToothSeg and Instance Grouping. ToothSeg [4] addresses panoptic tooth instance segmentation and numbering, while this paper reports semantic tooth labels with DSC. To isolate whether the remaining gap is due only to instance grouping, we used the official TeethSeg code to fuse nnU-Net predictions with ground-truth instances. The remaining gap in table S4 indicates that perfect instance grouping does not resolve cases where an ambiguous tooth is assigned the wrong semantic label; those cases require anatomical context beyond local boundaries.

Method	STS2024 DSC
nnU-Net	92.69
nnU-Net + ground-truth instance grouping	93.67
Teeth2Point	95.46

Table S4. Comparison with ground-truth instance grouping on STS2024.

C Robustness Analyses

Missing-Tooth Cases. The reviewers asked whether missing teeth were analyzed separately. Table S5 shows that missing-tooth cases are substantially harder than clean cases. Teeth2Point improves DSC by 1.5 points on missing-tooth cases and by 0.4 points on clean cases. Molar-related missing teeth account for approximately 76% of all missing teeth; on third molars, which account for approximately 52% of missing teeth, the gain reaches up to 3.1 DSC points over the baseline.

Method	Clean DSC (32%)	Missing-Tooth DSC (68%)
nnU-Net	93.8	89.1
Teeth2Point	94.2	90.6

Table S5. Performance stratified by clean and missing-tooth cases.

Case-Wise Variability. Because the workshop paper reports a single split, we additionally report case-wise mean and standard deviation over the full test set in table S6. The point-stage method increases the mean and lowers the case-wise standard deviation for both DSC and molar DSC.

Method	DSC	Molar DSC
nnU-Net	90.78 ± 4.13	88.56 ± 5.00
Teeth2Point	91.75 ± 3.45	90.25 ± 2.93

Table S6. Case-wise mean and standard deviation over the full test set.

Sensitivity to First-Stage ROI Errors. The second stage is bounded by first-stage foreground extraction: if a tooth is completely absent from the predicted ROI, the point network cannot recover it. We therefore stratified performance by the number of teeth missed by stage 1. The majority of cases have no missed teeth during ROI extraction, while severe misses remain the main failure mode.

Stage-1 Detection Status	Case Ratio	Stage-2 DSC
Missing ≥ 2 teeth	1.7%	75.24
Missing $= 1$ tooth	5.9%	87.33
Missing $= 0$ teeth	92.4%	92.22

Table S7. Second-stage performance stratified by first-stage missed-tooth count.

D Beyond-Teeth Analysis

The jawbone labels in ToothFairy-family data motivate a beyond-teeth test, but the point-stage design is intentionally optimized for sparse tooth foreground. When maxilla and mandible labels are included, jawbone points dominate the foreground token budget. In one representative instance, the encoder-stage point counts increase to $[768525, 108914, 17137, 3060, 591]$, so the stage with 3060 tokens exceeds the default attention window of 1024. Table S8 shows that adding jawbone labels lowers tooth DSC for VAP, suggesting that the current sparse point-stage configuration should remain tooth-focused unless a larger-window or class-balanced token strategy is introduced.

Method	Teeth		Molar/Maxilla		Premolar/Mandible	
	DSC	HD95	DSC	HD95	DSC	HD95
Ground-truth ROI						
VAP (teeth only)	93.66	12.47	92.32	18.16	93.31	14.46
VAP (teeth + jawbone)	90.63	12.22	89.70	14.08	89.86	18.27
Estimated ROI						
nnU-Net-M	88.08	19.40	88.68	12.14	96.41	2.94
VAP (teeth only)	88.79	13.04	88.68	12.14	96.41	2.94
VAP (teeth + jawbone)	87.91	17.63	87.62	11.09	95.83	4.43

Table S8. Additional analysis on maxilla and mandible labels. In the estimated-ROI block, the last four metric columns correspond to maxilla and mandible rather than molar and premolar.

E Additional Qualitative Results

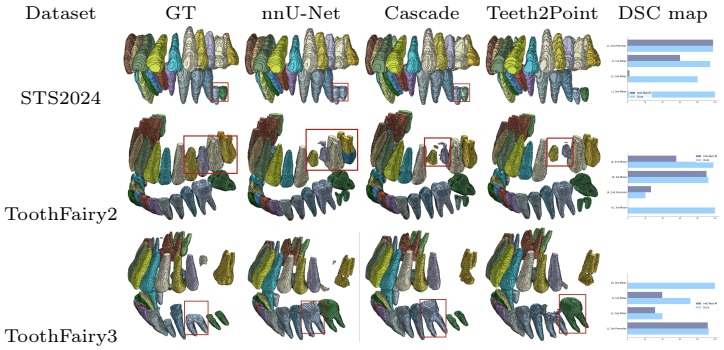


Fig. S1. Additional qualitative examples. The first-stage output provides the foreground support for both the cascade nnU-Net and Teeth2Point; when the first stage misses an entire tooth region, the second stage cannot recover that region.

F Reproducibility Notes

All supervised comparisons use identical 70/30 train-test splits within each dataset, identical post-processing, and case-wise metric aggregation. Hyperparameters that materially affect the point-stage behavior are reported in the main paper and expanded here: coordinate discretization $\epsilon = 2$, gradient-histogram bin width $\alpha = 0.125$, prediction voxelization $K = 4$, point-stage batch size 8, 400 GT-ROI training epochs, 100 predicted-ROI adaptation epochs, 200 self-supervised pretraining epochs, and the pretraining masking schedule from 0.3 to 0.7. Code release remains subject to internal intellectual-property and commercialization review.

References

1. Caron, M., et al.: Unsupervised learning of visual features by contrasting cluster assignments (2021)
2. Chen, X., He, K.: Exploring simple siamese representation learning (2020)
3. Isensee, F., et al.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods **18**(2), 203–211 (2021)
4. van Nistelrooij, N., et al.: Toothseg: Robust tooth instance segmentation and numbering in cbct using deep learning and self-correction. IEEE Journal of Biomedical and Health Informatics (2025). https://doi.org/10.1109/JBHI.2025.3650444
5. Oquab, M., et al.: Dinov2: Learning robust visual features without supervision (2024)
6. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015)
7. Roy, S., et al.: Mednext: Transformer-driven scaling of convnets for medical image segmentation (2024)
8. Wu, X., et al.: Point transformer v3: Simpler, faster, stronger. In: CVPR (2024)
9. Zhou, J., et al.: ibot: Image bert pre-training with online tokenizer (2022)