U-Mamba2-SSL for Semi-Supervised Tooth and Pulp Segmentation in CBCT

Zhi Qin Tan[1[0000−0002−5521−6808]], Xiatian Zhu[2[0000−0002−9284−2955]], Owen Addison[1[0000−0002−0981−687X]], and Yunpeng Li[1[0000−0003−4798−541X]]

[1] Centre for Oral, Clinical & Translational Sciences, King's College London, United Kingdom
{zhi_qin.tan,owen.addison,yunpeng.li}@kcl.ac.uk
[2] Surrey Institute for People-Centred AI, University of Surrey, United Kingdom
{xiatian.zhu,yunpeng.li}@surrey.ac.uk

Abstract. Accurate segmentation of teeth and pulp in Cone-Beam Computed Tomography (CBCT) is vital for clinical applications like treatment planning and diagnosis. However, manual segmentation requires extensive expertise and is exceptionally time-consuming, highlighting the critical need for automated semi-supervised segmentation algorithms that can utilize unlabeled data. In this paper, we propose **U-Mamba2-SSL**, a novel semi-supervised learning framework that builds on the U-Mamba2 model and employs a multi-stage training strategy. The framework first pre-trains U-Mamba2 in a self-supervised manner using a disruptive autoencoder. It then leverages unlabeled data through consistency regularization, where we introduce input and feature perturbations to ensure stable model outputs. Finally, a pseudo-labeling strategy is implemented with a reduced loss weighting to minimize the impact of potential errors. U-Mamba2-SSL obtained 0.917 DSC and 0.948 mIoU on the hidden test set, achieving first place in Task 1 of the STSR 2025 challenge. The code is available at https://github.com/zhiqin1998/UMamba2.

Keywords: Semi-supervised learning · U-Mamba2-SSL · CBCT Imaging · Tooth and Pulp Segmentation · STSR 2025 Challenge

1 Introduction

Cone-Beam Computed Tomography (CBCT) provides comprehensive 3D information of the oral region and is an important imaging tool in dentistry, as shown by its rapid adoption in dental clinics [10]. Precision segmentation of the tooth and pulp structures is vital to various applications such as dental conditions diagnosis, orthodontic procedures, treatment and surgery planning [14,20]. However, manual segmentation of CBCT scans requires specialized training and is extremely time-consuming due to its high resolution containing a massive number of voxels and the high variability across scans, making it impractical to scale up in practice. This highlights the significance of developing effective semi-supervised approaches with only limited labeled data while leveraging a large amount of unlabeled CBCT scans [2, 23, 24].

Semi-supervised learning (SSL) incorporates elements from both supervised and unsupervised learning [4,26], utilizing both labeled and unlabeled data to improve the performance on the supervised task by exploring the latent knowledge from unlabeled data. This alleviates the need for a significant amount of labels, which can require considerable resources to obtain. We focus on three categories of SSL: 1) Knowledge transfer with pre-training refers to the transfer of knowledge from one task to another via pre-training, where autoencoders [8, 21, 22] are trained to reconstruct corrupted input from a large amount of unlabeled data to guide the randomly initialized model weights towards potentially better regions; 2) Consistency regularization training [5, 13, 15] based on the smoothness assumption, enforces the model to produce similar output after perturbing the input, internal features, or model weights, pushing the model towards better generalization capability; and 3) Pseudo labeling method [12], one of the most common approaches in SSL due to its simplicity and model-agnostic nature. It is a form of entropy regularization [6] with unlabeled data, reducing the overlap of class probability distribution and favoring a low-density class separation.

In this paper, we present U-Mamba2-SSL, a multi-stage semi-supervised learning framework for tooth and pulp segmentation in 3D CBCT images, developed in the scope of the STSR 2025 Task 1 Challenge [1]. To exploit the vast amount of unlabeled CBCT data, we first pre-train U-Mamba2 [17] with the disruptive autoencoder on all provided data. Then, the second training stage involves using the labeled data for supervised learning and the unlabeled data for unsupervised learning via consistency regularization techniques in the input and feature spaces. Lastly, the final stage introduces the pseudo labeling method to the training procedure of the previous stage, with a lower loss weight to further optimize the model weights. The extensive experiments demonstrate the superior performance of our method, outperforming other alternatives and achieving first place with an average score of 0.789 in the STSR 2025 hidden test set.

2 Method

Fig. 1 shows the overall process of the U-Mamba2-SSL framework, consisting of three training stages where we first pre-train the U-Mamba2 [17] model with reconstruction objectives, then combine supervised loss for the labeled data and unsupervised loss with consistency regularization for the unlabeled data. The final third stage introduces pseudo labeling to the training objectives.

U-Mamba2 integrates Mamba2 [3] state space models into the U-Net architecture at the bottleneck region to enhance its ability to capture long-range dependencies. Mamba2 improves upon Mamba [7] by enforcing stronger constraints on the hidden space structure, leading to higher efficiency without compromising its performance compared to transformer-based alternatives. We present the details of the three training stages: pre-training, consistency regularization training, and pseudo labeling, in the following subsections. Note that the final checkpoint of each training stage is used to initialize the model of the subsequent stage.

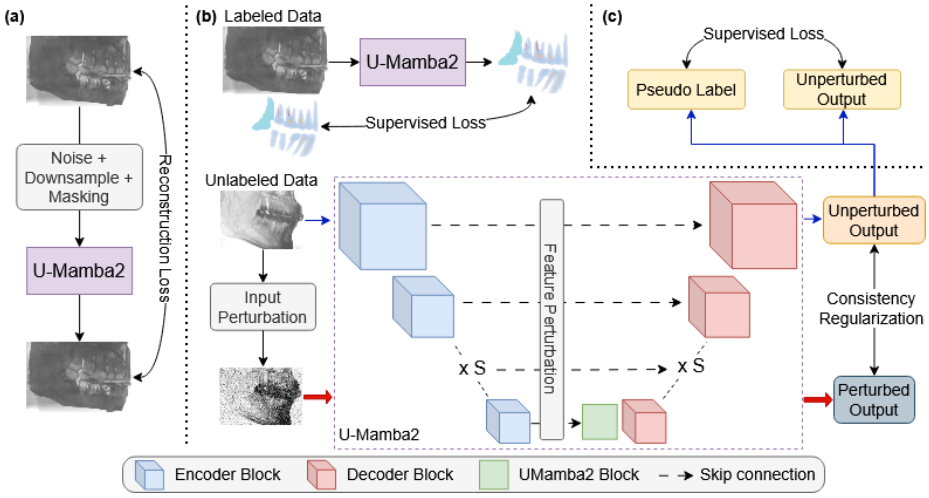


Fig. 1: Overall diagram of the proposed U-Mamba2-SSL framework. (a) The first pre-training stage; (b) The second consistency regularization training stage; (c) The third pseudo labeling training stage.

Problem Formulation. Let $\mathcal{D}_l = \{(x_1^l, y_1), ..., (x_n^l, y_n)\}$ represent the n labeled samples and $\mathcal{D}_u = \{x_1^u, ..., x_m^u\}$ represent the m unlabeled samples, where $x_i^l \in \mathbb{R}^{H \times W \times D}$ is the i-th labeled input image, $y_i \in \mathbb{R}^{C \times H \times W \times D}$ is its corresponding voxel-level label, and x_i^u is the i-th unlabeled input image. Here, C is the number of classes while H, W, D are the spatial dimensions. Our goal is to exploit the larger number of unlabeled samples ($i.e.$ $m \gg n$) to train a 3D segmentation model.

2.1 First Stage: Pre-training with Disruptive Autoencoder

In the medical image domain, data scarcity due to various factors such as complex ethical regulations for accessing and releasing datasets publicly, presents challenges to model pre-training. Therefore, unlike in computer vision tasks of natural images, models for medical image applications are often trained from scratch with random initialization of model weights. However, recent works [18, 21] have shown that pre-training deep learning models for medical image tasks can lead to better models that can extract meaningful feature representations to enhance the performance of downstream segmentation tasks, particularly when there is limited labeled data to train from scratch effectively.

In the first stage of our proposed SSL framework, we utilize all training data ($i.e.$ $\mathcal{D}_l \cup \mathcal{D}_u$) to pre-train U-Mamba2 via the disruptive autoencoder (DAE) [21] method. The DAE method combines three low-level reconstruction tasks for pre-training, namely denoising, super-resolution, and recovering masked information.

Denoising refers to the task of restoring the original input from its noisy version, obtained by introducing random additive Gaussian noise to the original

input. The model must learn to restore all local details in images, such as edges and textures, to output a good denoised image. Besides that, super-resolution is the task of increasing the resolution of a low-resolution image, created artificially by downsampling the original input with linear interpolation. To obtain a good upsampled image, the model must be able to recover the fine details of the image with both local and global information. Lastly, we apply masking to random cubical regions in the input image, setting the voxel values to zero. As most of the information in medical images is not global but is in the finer local details, we use a small cube size relative to the spatial dimensions of the input to prevent discarding too much local information. The model is directed to recover the masked regions, leading to the ability to extract meaningful global context. After applying the three input disruptions, U-Mamba2 learns to reconstruct the original image from the corrupted input with an L1 loss function.

2.2 Second Stage: Consistency Regularization Training

We exploit the smoothness assumption and employ consistency regularization training in the second training stage, enforcing the invariance of predictions on the model. In this training stage, we use a combination of supervised loss and unsupervised loss to learn the model parameters. For a labeled training sample, x_i^l, and its voxel-level class label, y_i, the model is trained in a supervised fashion based on the combination of Dice loss and cross-entropy loss, $\mathcal{L}_S$. For an unlabeled training sample, x_i^u, it is first passed through the model to obtain an unperturbed output, $\hat{y}_i^u$. Then, we introduce input and feature perturbations [13] to x_i^u and obtain the perturbed output, $\tilde{y}_i^u$, by passing the perturbed input through the model. The semi-supervised consistency regularization loss, $\mathcal{L}_{CR}$, is computed as the $\mathcal{L}_1$ loss between $\hat{y}_i^u$ and $\tilde{y}_i^u$. We describe the perturbation details in the following paragraphs.

Input Perturbations. We apply strong data augmentation to the unlabeled data to obtain a perturbed input. It is crucial not to apply spatial (*e.g.* mirroring or rotation) augmentations, as in the context of segmentation, these transformations are non-local and violate the smoothness assumption. Specifically, in this stage, we apply median filter, Gaussian blur, Gaussian noise, random brightness, random contrast, low-resolution simulation, and image sharpening filter.

Feature Perturbations. The perturbed inputs are passed through the encoder blocks in U-Mamba2 to obtain multi-scale 3D feature maps. Before the encoder feature maps are connected to the decoder blocks via skip connections, we apply random perturbations in the feature space to encourage the model to learn more robust and generalizable feature representations. The feature perturbations consist of dropping activations or injecting noise in the encoder feature maps:

- Random Spatial Dropout [19]: We apply random channel-wise dropout with a probability of 0.5. In contrast to i.i.d. dropout, this promotes channel-wise independence in the encoder feature maps.

- Random Activation Dropout [16]: Activations with high values are randomly dropped to enforce the model to focus on inactive regions in the feature map. We randomly sample a threshold, $\gamma_{drop} \sim \mathcal{U}(0.7, 0.9)$, then set all activations above the γ_{drop} percentile to zero. As a result, the top $10\% - 30\%$ highly activated regions in the feature map are dropped.
- Noise Injection: A noise tensor with the same shape as the feature map is first sampled from a uniform distribution, $N \sim \mathcal{U}(-0.3, 0.3)$. As the activations in the feature maps vary, we ensure that the noise tensor is proportional to the feature map by first multiplying the noise tensor with the feature map before adding it as $Z + (Z \odot N)$, where $Z \in \mathbb{R}^{F \times H \times W \times D}$ is the feature map, $\odot$ is element-wise multiplication, and F is the number of channels.

Semi-Supervised Learning Schedule. In practice, we utilize both labeled and unlabeled data during each training epoch. The overall loss signal from both labeled and unlabeled data is computed as

$$\mathcal{L} = \mathcal{L}_S + \omega_{CR}\mathcal{L}_{CR} \ , \tag{1}$$

where ω_{CR} is the unsupervised loss weight function. ω_{CR} ramps up exponentially [11] from zero to a fixed weight, W_{CR}, at the $0.2T_{ep}$ epoch where T_{ep} is the total number of training epochs. Additionally, we linearly increase the proportion of unlabeled data in each epoch from 10% to 50% at the $0.4T_{ep}$ epoch, allowing the model to focus on learning the main segmentation task in the early phase.

2.3 Third Stage: Pseudo Labeling

After the second training stage, we obtain a good U-Mamba2 segmentation model that can maintain local smoothness around its predictions. We capitalize on this feature by further training the model with the pseudo labeling [12] strategy. Specifically, the model's predictions on unlabeled samples are considered pseudo labels and used for model training in a supervised manner. For the predicted class of each voxel, if the class confidence is above a given confidence threshold, λ_{conf}, then we use the predicted class as ground truth; otherwise, the voxel is set to the background class and is ignored in the loss calculation.

In this stage, the loss function from Eq. (1) becomes:

$$\mathcal{L} = \mathcal{L}_S + \omega_{CR}\mathcal{L}_{CR} + W_{PL}\mathcal{L}_{PL} \ , \tag{2}$$

where $\mathcal{L}_{PL}$ is the supervised loss computed with the pseudo labels and ignores the background class, and W_{PL} is the loss weight for $\mathcal{L}_{PL}$ to balance the loss terms. Similar to the second stage, we linearly increase the proportion of unlabeled samples in each training epoch from 30% to 50% at the $0.2T_{ep}$ epoch.

3 Experiments

3.1 Dataset and Evaluation Metrics

The evaluation metrics include Dice Similarity Coefficient (DSC), Normalized Surface Distance (NSD), Mean Intersection over Union (mIoU), and Identifica-

Table 1: Development environments and requirements.

System	Ubuntu 24.04
CPU	Intel(R) Core(TM) Ultra 9 285K
RAM	2 × 32GB; 6400 MHz
GPU	NVIDIA RTX 5090 32 GB
CUDA version	12.9
Programming language	Python 3.11
Deep learning framework	PyTorch 2.7.1, nnU-Net 2.6.2

tion Accuracy (IA) to evaluate the segmentation region overlap and boundary distance. In addition, the algorithm runtime and memory consumption are also evaluated and ranked.

3.2 Implementation Details

Preprocessing. We resize all inputs to the median voxel spacing of all training data, $(0.3, 0.25, 0.25)$, resulting in a median input size of $(337, 640, 640)$. Then, we clip the input data to the 0.5th and 99.5th percentiles, followed by data normalization based on the mean and standard deviation of the voxel values.

Environment settings. The development environments and requirements are presented in Table 1.

Training protocols. We implement U-Mamba2-SSL with the nnU-Net [9] framework, using a patch-size training and sliding window inference strategy. During training, we randomly apply rotation, scaling, Gaussian noise, Gaussian blur, brightness and contrast transform, low resolution simulation, and mirroring as data augmentation. We randomly crop input patches so that at least 33% of the voxels contain a foreground label. W_{CR}, W_{PL}, and λ_{conf} are set to 50, 0.1, and 0.75, respectively. All models have 7 encoder-decoder stages and follow the model configuration in Table 2. The provided 30 labeled training samples are split into 20 training and 10 internal validation splits, where the internal validation split is used to monitor training progress and offline evaluation. We select the checkpoint with the highest DSC on our internal validation set and report the performance metrics on the hidden validation set.

4 Results and Discussion

4.1 Quantitative Results

Table 3 presents the results of our proposed method compared with two baselines, nnU-Net and U-Mamba2. We observe that all methods achieved high DSC, NSD, and mIoU metrics, which measure overall image-level performance. However, U-Mamba2-SSL outperforms others significantly in IA, which calculates the average percentage of classes with IoU > 0.5 across all images. The bottom three rows

Table 2: Training configuration.

Pre-trained Model	See Section 2.1
Batch size	2
Patch size	$128 \times 256 \times 256$
Total epochs	500
Optimizer	SGD with 0.99 momentum
Initial learning rate	0.01
Lr decay schedule	Polynomial LR decay
Training time	13 hours
Loss function	See Equations (1) and (2)
Number of model parameters	156M
Number of flops	6.22T

Table 3: Evaluation results on the validation set. CR denotes consistency regularization; PL denotes pseudo label. Our ablation study is reported in the bottom three rows, with the last row referring to the final U-Mamba2-SSL.

Methods	Pre-train	CR	PL	DSC	NSD	mIoU	IA	Average
nnU-Net [9]	-	-	-	0.963	0.997	0.928	0.286	0.794
U-Mamba2 [17]	-	-	-	0.965	0.998	0.930	0.464	0.839
U-Mamba2-SSL	✓	×	×	0.967	0.998	0.937	0.731	0.908
	✓	✓	×	0.967	0.999	0.935	0.736	**0.910**
	✓	✓	✓	0.967	0.999	0.935	0.738	**0.910**

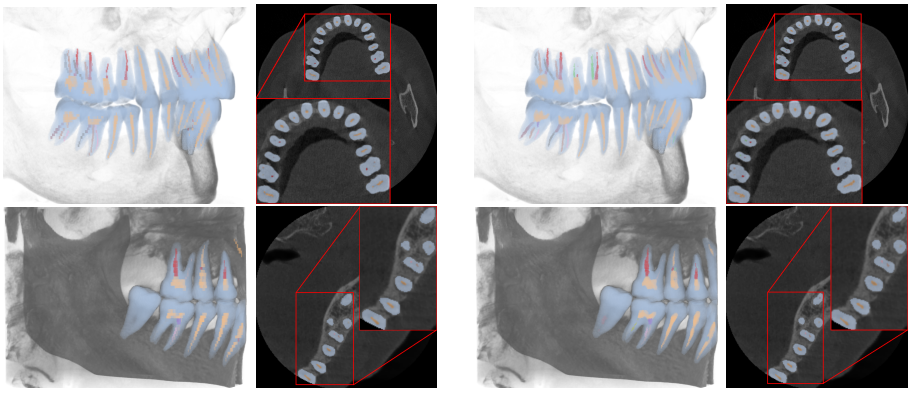
of Table 3 also report the ablation study of our proposed method. Notably, pre-training leads to the largest leap in IA, from 0.464 to 0.731, while incorporating consistency regularization and pseudo labeling further increases IA to 0.738.

4.2 Qualitative Results

Fig. 2 shows the qualitative comparison between the ground truth and our model's predictions of the scans with the highest and lowest DSC in our internal validation set, in the top and bottom rows, respectively. Generally, we observe that our method can accurately differentiate between the tooth and different classes of pulp and root canal. The failure cases of our method typically stem from the inability to precisely predict the thickness and the length or extent of the pulp. Moreover, our model also struggles with limited field of view (LFOV) CBCTs where it predicts more false positives around the image edges.

4.3 Final Challenge Submission

We scale up our training procedure by training on all available data for 1000 epochs and increasing the input patch size to 160x256x256. For inference, we use a sliding window inference with a tile size of 0.9, and enable mirroring in the anterior/posterior and left/right axes during test-time augmentation (See Appendix A for the speed optimization). Our method achieved a 0.969 DSC,



(a) Ground Truth (b) Prediction

Fig. 2: Qualitative results of U-Mamba2-SSL on the internal validation set. The 3D render and a representative 2D slice are shown for: (Top) the best scoring case and (Bottom) the worst scoring case.

0.998 NSD, 0.940 mIoU, and 0.806 IA on the validation set, while obtaining a DSC, NSD, mIoU, and IA of 0.917, 0.882, 0.948, and 0.577, respectively, on the final hidden test set, securing first place in Task 1 of the STSR 2025 challenge.

4.4 Limitation and Future Work

Our work, while successful, is not without limitations. First, the dataset consists of full and LFOV CBCTs, which differ in content and image properties. Next, the IA metric drops significantly on the final hidden test set, signifying possible overfitting or domain shift. Future work should design data processing and augmentation techniques tailored to the different types of CBCTs to leverage their differences and improve model generalizability. Lastly, as only a small region of interest (ROI) in the CBCT image contains the foreground classes, future research can exploit this to prevent wasting computation on non-foreground regions, allowing the model to focus on the true ROI.

5 Conclusion

We presented U-Mamba2-SSL, a novel multi-stage semi-supervised learning framework for tooth and pulp segmentation in CBCT scans, in the scope of the STSR 2025 challenge. The framework consists of first pre-training U-Mamba2 with the disruptive autoencoder, utilizing unlabeled data for consistency regularization, and a pseudo labeling strategy in the final stage. Our results demonstrate that the proposed framework can substantially enhance model performance, achieving first place with an average score of 0.789 on the hidden test set in Task 1 of the STSR 2025 challenge.

Acknowledgements We thank all the data owners for making the medical images publicly available and Codabench [25] for hosting the challenge platform.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. MICCAI STSR 2025 Challenge Task 1: Semi-supervised Teeth and Plup Root Canal Segmentation in 3D CBCT Scans. https://www.codabench.org/competitions/6468, accessed: 25th August 2025
2. Cui, W., Wang, Y., Li, Y., Song, D., Zuo, X., Wang, J., Zhang, Y., Zhou, H., Chong, B.s., Zeng, L., Zhang, Q.: Ctooth+: A large-scale dental cone beam computed tomography dataset and benchmark for tooth volume segmentation. In: MICCAI Workshop on Data Augmentation, Labelling, and Imperfections. pp. 64–73 (2022)
3. Dao, T., Gu, A.: Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In: Proc. Int. Conf. Mach. Learn. (ICML). Vienna, Austria (2024)
4. van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Machine Learning **109**(2), 373–440 (Feb 2020)
5. Fan, Y., Kukleva, A., Dai, D., Schiele, B.: Revisiting consistency regularization for semi-supervised learning. Int. J. Comput. Vis. (IJCV) **131**(3), 626–643 (Mar 2023)
6. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Proc. Adv. Neural Inform. Process. Syst. (NeurIPS) (2004)
7. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv:2312.00752 (2023)
8. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 16000–16009 (2022)
9. Isensee, F., Jaeger, P.F., Kohl, S.A.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature Methods **18**(2), 203–211 (Feb 2021)
10. Jaju, P.P., Jaju, S.P.: Clinical utility of dental cone-beam computed tomography: current perspectives. Clin. Cosmet. Investig. Dent. **6**, 29–43 (2014)
11. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv:1610.02242 (2016)
12. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Int. Conf. Mach. Learn. (ICML) Workshop: Challenges in Representation Learning (WREPL) (2013)
13. Ouali, Y., Hudelot, C., Tami, M.: Semi-supervised semantic segmentation with cross-consistency training. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2020)
14. Patel, S., Durack, C., Abella, F., Shemesh, H., Roig, M., Lemberg, K.: Cone beam computed tomography in endodontics – a review. International Endodontic Journal **48**(1), 3–15 (2015)
15. Sinha, S., Dieng, A.B.: Consistency regularization for variational auto-encoders. In: Proc. Adv. Neural Inform. Process. Syst. (NeurIPS). pp. 12943–12954 (2021)

16. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. (JMLR) **15**(56), 1929–1958 (2014)
17. Tan, Z.Q., Zhu, X., Addison, O., Li, Y.: U-mamba2: Scaling state space models for dental anatomy segmentation in cbct. arXiv:2509.12069 (2025)
18. Tang, Y., Yang, D., Li, W., Roth, H.R., Landman, B.A., Xu, D., Nath, V., Hatamizadeh, A.: Self-supervised pre-training of swin transformers for 3d medical image analysis. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 20698–20708. New Orleans, LA, USA (2022)
19. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 648–656 (2015)
20. Tyndall, D.A., Price, J.B., Tetradis, S., Ganz, S.D., Hildebolt, C., Scarfe, W.C.: Position statement of the american academy of oral and maxillofacial radiology on selection criteria for the use of radiology in dental implantology with emphasis on cone beam computed tomography. Oral Surg Oral Med Oral Pathol Oral Radiol **113**(6), 817–826 (2012)
21. Valanarasu, J.M.J., Tang, Y., Yang, D., Xu, Z., Zhao, C., Li, W., Patel, V.M., Landman, B.A., Xu, D., He, Y., Nath, V.: Disruptive autoencoders: Leveraging low-level features for 3d medical image pre-training. In: Proc. Int. Conf. on Med. Imag. with Deep Learning. vol. 250, pp. 1553–1570 (2024)
22. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. (JMLR) **11**(110), 3371–3408 (2010)
23. Wang, Y., Ye, F., Chen, Y., Wang, C., Wu, C., Xu, F., Ma, Z., Liu, Y., Zhang, Y., Cao, M., Chen, X.: A multi-modal dental dataset for semi-supervised deep learning image segmentation. Scientific Data **12**(1), 117 (Jan 2025)
24. Wang, Y., Zhang, Y., Chen, X., Wang, S., Qian, D., Ye, F., Xu, F., Zhang, H., Dan, R., Zhang, Q., et al.: MICCAI 2023 STS challenge: A retrospective study of semi-supervised approaches for teeth segmentation. Pattern Recognition **170**, 112049 (2026)
25. Xu, Z., Escalera, S., Pavão, A., Richard, M., Tu, W.W., Yao, Q., Zhao, H., Guyon, I.: Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. Patterns **3**(7), 100543 (2022)
26. Yang, X., Song, Z., King, I., Xu, Z.: A survey on deep semi-supervised learning. IEEE Trans. on Knowledge and Data Engineering **35**(9), 8934–8954 (2023)

Supplementary Material

A Optimizing Speed in Sliding Window Inference

Since inference time is also an evaluation metric in the STSR 2025 challenge, we explore optimizing the parameters of the sliding window inference technique to improve inference speed without significantly deteriorating model performance (*i.e.* the average score of DSC, NSD, mIoU, and IA). Fig. 3 illustrates the tradeoff between the model performance and inference time for different tile sizes and mirror axes combinations in test-time augmentation (TTA). As most of the voxels in the CBCT image belong to the background class, setting the tile size to 0.9 substantially reduces the inference time by 53% with a negligible drop of only 0.002 average score. Furthermore, Fig. 3 demonstrates that although mirroring in all axes leads to the best performance, it comes with the downside of long inference time. The optimal mirror axes combination is '1,2', offering a good average score with an inference time of only 17.08 seconds.

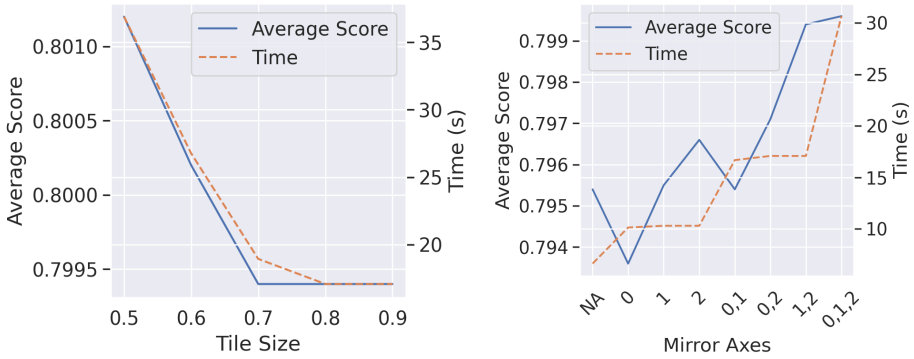


Fig. 3: (Left): Effect of the tile size on the metrics with '1,2' mirror axes in TTA. (Right): Effect of various mirror axes combinations in TTA on the metrics when tile size is set to 0.9. Axis definition: '0' is superior/inferior, '1' is anterior/posterior, and '2' is left/right.