Disconnect to Connect: A Data Augmentation Method for Improving Topology Accuracy in Image Segmentation

Juan Miguel Valverde[1,2] JMVMA@DTU.DK
Maja Østergaard[3] MAJAOESTER@INANO.AU.DK
Adrian Rodriguez-Palomo[3] ADRIAN.RODRIGUEZ@INANO.AU.DK
Peter A. S. Vibe[3] PETERVIBE@INANO.AU.DK
Nina K. Wittig[3] NKW@INANO.AU.DK
Henrik Birkedal[3] HBIRKEDAL@CHEM.AU.DK
Anders Bjorholm Dahl[1] ABDA@DTU.DK

[1] *DTU Compute, Technical University of Denmark, Denmark*
[2] *A.I. Virtanen Institute, University of Eastern Finland, Finland*
[3] *Department of Chemistry and iNANO, Aarhus University, Denmark*

Editors: Under Review for MIDL 2026

Abstract

Accurate segmentation of thin, tubular structures remains a significant challenge for deep neural networks, where minor misclassifications can lead to broken connections and topologically incorrect results. Current approaches for improving topology accuracy, such as topology loss functions, require training labels that are both precise and topologically accurate. However, accurate labels are hard to obtain since image annotation is laborious, time-consuming, and the images' low resolution and low contrast can make tubular structures appear disconnected. We present CoLeTra, a data augmentation method that improves topology accuracy, even on datasets with topologically inaccurate training labels. CoLeTra achieves this by artificially breaking the structures in the images while maintaining the original labels, teaching models that structures can be continuous even if they appear disconnected. We evaluated CoLeTra on three datasets with six loss functions and two architectures, demonstrating that CoLeTra generally improved topology accuracy while often improving the Dice coefficient and clDice. We also release a dataset specifically suited for image segmentation methods with a focus on topology accuracy. CoLetra's code can be found at https://github.com/jmlipman/CoLeTra.

Keywords: Topology, Image Segmentation, Data Augmentation, Inpainting

1. Introduction

Despite the remarkable potential of deep learning for image segmentation (Minaee et al., 2021), accurate segmentation of thin, tubular structures, such as axons, airways, and blood vessels, remains a significant challenge. This is partly due to the inherent pixel-wise classification nature of standard deep learning models, where small misclassifications that hardly increase the loss, can lead to segmentations with discontinuities, impairing the quantification (Hu et al., 2022). For instance, connected-component analysis of structures with broken connections would misinterpret each fragment as a distinct structure, resulting in overestimating their number. An incorrect number of connected components would then compromise other measurements, such as average length, density, and directionality.

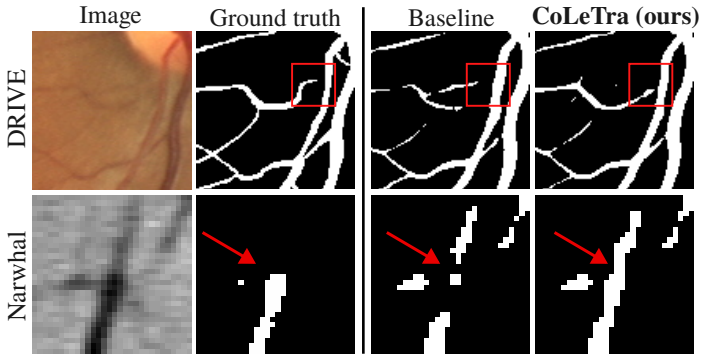

Figure 1: Left: Annotations missing important areas, compromising the structure's true topology. Right: CoLeTra improves topology accuracy.

An important line of research has focused on emphasizing structures' connectivity with topology loss functions (Hu et al., 2019; Clough et al., 2020; Hu et al., 2022; Liao, 2023; Shit et al., 2021). These are loss functions that aim to produce segmentations that match the number of connected components, holes, and cavities of the ground truth. However, the effectiveness of topology loss functions requires the ground truth to be topologically precise, limiting the usability of topology loss functions on datasets where topology accuracy cannot be guaranteed. This limitation is common in medical and material science images (Shi et al., 2024a), especially in large 3D images, since these images are difficult, time-consuming to annotate, and their low resolution and contrast can make structures to appear disconnected, further reducing the quality and topology accuracy of the annotations (see Figure 1). In these cases, with topologically inaccurate labels, the loss functions inadvertently reinforce the absence of a structure that we know should exist based on prior knowledge. Furthermore, most topology loss functions rely on persistence homology (Oner et al., 2023; Stucki et al., 2023; Xu et al., 2024; Zhang et al., 2025), soft-skeletonization (Shit et al., 2021; Viti et al., 2022), or distance transforms (Shi et al., 2024b; Liu et al., 2024), which significantly increase the computational resources required by topology losses, limiting their applicability.

In this paper, we present CoLeTra, a data augmentation method that breaks the structures in the training set images while keeping the ground truth labels as they are, teaching models that structures may be connected even if they appear disconnected. Our extensive experiments, including two architectures and six loss functions, show that CoLeTra improves topology accuracy on datasets with accurate labels and with topologically inaccurate labels. CoLeTra adds only a negligible overhead during the optimization and it requires no extra GPU memory. Our contributions are:

- We present CoLeTra, the first, to the best of the authors knowledge, data augmentation method for image segmentation specifically designed to and demonstrated to improve topology accuracy.
- CoLeTra improves topology accuracy across different settings, even when optimizing topology loss functions.

- We release a dataset suited for evaluating image segmentation methods focused on topology accuracy. This dataset release aims to foster future research in the field.

2. Related work

Topology loss functions At the core of several topology loss functions, we find persistence homology (PH) (Edelsbrunner et al., 2002)—a tool to analyze topological features. Particularly in image segmentation, PH tracks how connected components, holes, and cavities appear and disappear as all possible thresholds are applied to softmax probability values (Hofer et al., 2017). Hu et al. (2019) developed a loss function that uses PH to find topological features and encourage some of them to be preserved while others to be removed. Gabrielsson et al. (2020) introduced a differentiable topology layer to incorporate topological priors in deep neural networks. Clough et al. (2020) and Shin et al. (2020) used PH to infuse a pre-specified type and number of topological features during training and at inference time. Byrne et al. (2022) and He et al. (2023) extended PH losses to multi-class segmentation, and Oner et al. (2023) employed a novel filtration technique, utilized by PH to identify topological features, that combines approaches from topological data analysis. Stucki et al. (2023) proposed a method to spatially match features between the segmentation's and label's PH barcodes. Although there are libraries that facilitate the computation of PH, such as Gudhi (Maria et al., 2014), computing PH on large images is very expensive. This issue has forced researchers to compute PH on small image patches, thus, disregarding global topology. Our method, in contrast, increases computational time only by a small margin and, as we show, it improves global topology accuracy.

In parallel to this trend, other topology loss functions do not rely on PH. Hu (2022) proposed to correct the critical points, *i.e.*, the pixels that change the topology of the segmentation to the desired one. This method, however, requires computing distance maps several times, slowing down the training considerably. Liao (2023) developed a method based on Dijkstra's algorithm that is also reportedly expensive to compute. Shit et al. (2021) presented "centerline Dice loss" (clDice), which, through a novel differentiable skeletonization algorithm, focuses on achieving accurate skeletons of the region of interest. Although this method is faster to compute than the PH-based loss functions, it requires larger GPU memory. More recently, Shi et al. (2024b) and Kirchhoff et al. (2024) also presented topology loss functions that emphasize performance in the centerline area. In contrast to these approaches, our method does not require extra GPU memory and it does not rely on computationally-expensive distance transform functions.

Prior knowledge via data augmentation Data augmentation can integrate prior knowledge into the models during training by leveraging domain-specific knowledge. In one example, the medical knowledge that tumors can grow in different locations has been incorporated by randomly flipping the images during the optimization (Myronenko, 2019). In another example, intra and inter-rater variability was simulated by applying random deformations to the segmentation mask boundaries (Javaid et al., 2019). In other domains, Yang et al. (2021) proposed a data augmentation transformation to create defects in metal surfaces accounting for new defect types that only appeared in the test set.

Inpainting to increase generalizability Image inpainting is used for reconstructing missing parts of an image, *e.g.*, to replace a removed object. However, a range of other

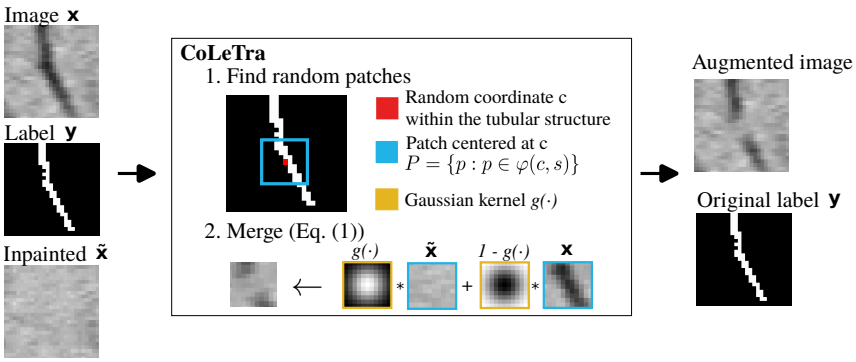

Figure 2: Overview of CoLeTra data augmentation.

applications have emerged. Inpainting has been used as a self-supervised learning task to achieve models with better feature representations. Pathak et al. (2016) and Zhong et al. (2020) trained models to reconstruct areas filled with random values; Chen et al. (2019) reconstructed images with swapped random patches; and Li et al. (2021) focused on reconstructing the most discriminative regions. Another line of research employs image inpainting to achieve larger, more diverse training sets. These approaches use off-the-shelf image inpainting methods to create "background images", where objects have been removed, for later to place other objects on those background images. Ruiz et al. (2019, 2020) proposed a method that decides with k-nearest neighbors which object to put in the inpainted background images. Zhang et al. (2020) trained a model named PlaceNet that predicts potential object locations and scales within the background images. Wang et al. (2022) developed a method that determines objects' location via a genetic algorithm. Nie et al. (2022) focused on creating synthetic datasets for pedestrian detection. Saha et al. (2021) approach detects vehicles with a Mask RCNN (He et al., 2017) and removes them via inpainting. More recently, He et al. (2024) used image inpainting to generate images with objects from the minority class to tackle class imbalance. In this paper, we use image inpainting to augment data with appearance of change in topology.

3. Method

In this section, we present our **Co**ntinuity **Le**arning **Tra**nsformation (**CoLeTra**), a data augmentation method to improve topology accuracy. Let $\mathbf{x} \in \mathbb{R}^d$ be an image of d pixels from a training mini-batch, and let $\mathbf{y} \in \{0,1\}^d$ be its corresponding ground-truth segmentation mask. CoLeTra transforms $\mathbf{x}$ during training:

$$x_p \leftarrow g(p,c,\sigma)\tilde{x}_p + (1 - g(p,c,\sigma))x_p \qquad \forall p \in \varphi(c,s), \forall c \in C, \tag{1}$$

where $C = \{c_1, \ldots, c_n\}$ is a set of n pixel coordinates such that $y_{c_i} = 1$; $\tilde{\mathbf{x}}$ is an inpainted version of $\mathbf{x}$ with all the thin structures removed; $\varphi(c,s)$ is the set of pixel coordinates within a window size of $s \times s$ centered at c; and $g(p,c,\sigma) = \exp(-\frac{1}{2}\frac{||p-c||_2^2}{\sigma^2}) \in [0,1]$ is the value of a Gaussian kernel that weighs the contribution of $\tilde{\mathbf{x}}$ and $\mathbf{x}$ at pixel p (see Figure 2). For the sake of clarity, we simplified our notation; CoLeTra extends to multi-class problems, and, as

we show, to multi-channel n-dimensional images. Intuitively, CoLeTra erases random parts of the tubular structures via image inpainting (simulating low resolution/contrast) while leaving the ground truth as it is, teaching the models that structures may be connected even if they appear disconnected.

CoLeTra's strategy is based on two key observations: 1) the fragmentation of tubular structures is often driven by structures' thin width and images' low resolution and contrast, and 2) in the training labels, tubular structures are more likely to be incorrectly broken than incorrectly connected (*i.e.*, we can generally assume that the labels indicate entire structures and/or fragments). CoLeTra is agnostic to the inpainting method, allowing it to always utilize state-of-the-art inpainting methods and inpainting methods that are more suitable for specific datasets. In our implementation, we used LaMa (Suvorov et al., 2022) to remove all tubular structures. To ensure smooth borders and complete coverage of thin structures, the areas for inpainting were defined by dilating the labels three times:

$$\tilde{\mathbf{x}} = LaMa(\mathbf{x}, dilate(\mathbf{y}, times = 3)). \tag{2}$$

CoLeTra also uses a simple method for finding where to artificially break the thin structures. During training, CoLeTra generates a set of random pixels by sampling, with the same probability, n pixels containing the structure to be segmented:

$$C \subset \{i : y_i = 1\} : |C| = n. \tag{3}$$

CoLeTra, then, centers the inpainted patches in each pixel belonging to the subset C.

CoLeTra's simplicity provides two important benefits. First, CoLeTra only adds a negligible overhead (in the order of *ms* per iteration) to the training while requiring no extra GPU memory. Second, the strategy for finding where to disconnect structures can be tailored to each dataset based on prior knowledge. For instance, disconnections could be applied to areas with lower contrast or at points where structures bifurcate, prioritizing the correction of regions with these specific characteristics. In this work, we show in datasets with accurate and topologically inaccurate labels that, even with a simple strategy, CoLeTra achieves segmentations topologically more accurate.

4. Experiments

4.1. Benchmark

Datasets and data split We evaluated CoLeTra on three medical datasets: DRIVE (Staal et al., 2004), PulmonaryAV (Cheng et al., 2024), and the Narwhal dataset—that we make publicly available. The DRIVE dataset comprised 20 fundus retina images collected for diabetic retinopathy detection, and the segmentation task focused on blood vessel segmentation. Due to the small dataset size, we conducted three-fold cross validation, and we further applied a 0.8:0.2 split to yield the training and validation sets. The PulmonaryAV dataset included 106 3D CT scans of chest images that we divided into a 74-10-22 train-validation-test split. In this dataset, the task was binary segmentation of pulmonary arteries and veins. The Narwhal dataset comprised synchrotron 3D images from a narwhal tusk, where the task was to segment the dentine tubules (see details in Appendix A). The dataset consisted of 20 patches from a single scan for training, one patch from a different scan for

validation, and three patches from other scans for testing. Importantly, the training labels of DRIVE and PulmonaryAV are accurate, whereas the training labels of the Narwhal dataset are topologically incorrect since they were generated semi-automatically, resulting in thousands of broken structures. The test set labels were obtained differently to maximize topology accuracy, allowing to measure performance reliably (Appendix A).

Experimental setup We evaluated CoLeTra on 36 scenarios, combining two architectures, six loss functions, and the three datasets mentioned above (two datasets with topologically accurate training labels, and one with topologically inaccurate training labels). We trained a DynUNet, which is based on the state-of-the-art nnUNet(Isensee et al., 2021), and an AttentionUNet (Oktay et al., 2018), that incorporates attention-based modules. The loss functions that these models optimized were selected based on their characteristics. Cross Entropy (CE) and Dice loss (Milletari et al., 2016) are the most used loss functions for image segmentation; Dice loss, in addition, accounts for dataset imbalance. RegionWise loss (RWLoss) (Valverde and Tohka, 2023) is a loss function that incorporates pixel importance with distance maps and can improve topology accuracy (Liu et al., 2024). clDice (Shit et al., 2021), Warping (Hu, 2022), and Topoloss (Hu et al., 2019), are topology loss functions; clDice focuses on achieving accurate skeletons in tubular structures; Warping loss focuses on rectifying critical pixels that affect the topology of the prediction; and Topoloss focuses on correcting topological features leveraging persistence homology. For the DRIVE datasets, we trained 2D models, whereas for the PulmonaryAV and Narwhal datasets we trained 3D models. For the 36 experimental scenarios, we compared models trained with extensive data augmentation (baseline) against the same setup plus our CoLeTra. We repeated each experiment with three different random seeds. Our experiments ran on a cluster with NVIDIAs V100 (32GB), with each experiment taking between 1 hour and 24 hours depending on the dataset and loss function. Other details regarding the optimization can be found in Appendix B.

CoLeTra's hyper-parameters CoLeTra has two main hyper-parameters: the number of patches (n, *i.e.*, $|C|$ in Equation (1)) and patch size (s). Their optimal value, as in any data augmentation method, depends on the dataset and task. Since our goal was not to find their optimal value in each experimental scenario, we conducted a very simple strategy that explored only nine configurations per architecture and dataset. We optimized Cross Entropy loss with patches of size 11, 15, and 19—since the structures have a similar width—and a number of patches such that, after applying CoLeTra, the maximum covered area of the structures corresponds to approximately 40%, 50%, and 60%, avoiding breaking the structures too little or too much. The exact configurations that we explored and the configuration that yielded the best performance are detailed in Appendix C.

Metrics We measured the Dice coefficient, the centerline Dice (clDice) (Shit et al., 2021), and the Betti errors. We chose these metrics since they measure different characteristics of the segmentations at the pixel and topology level. The Betti errors specifically refer to the difference in the number of connected components (β_{0e}), holes (β_{1e}), and cavities (β_{2e}) between the predicted mask and the ground truth. These can be computed locally through several random (or sliding window) patches across the image (Lin et al., 2023), or globally in the entire image depending on whether local or global topology is more relevant. Due to

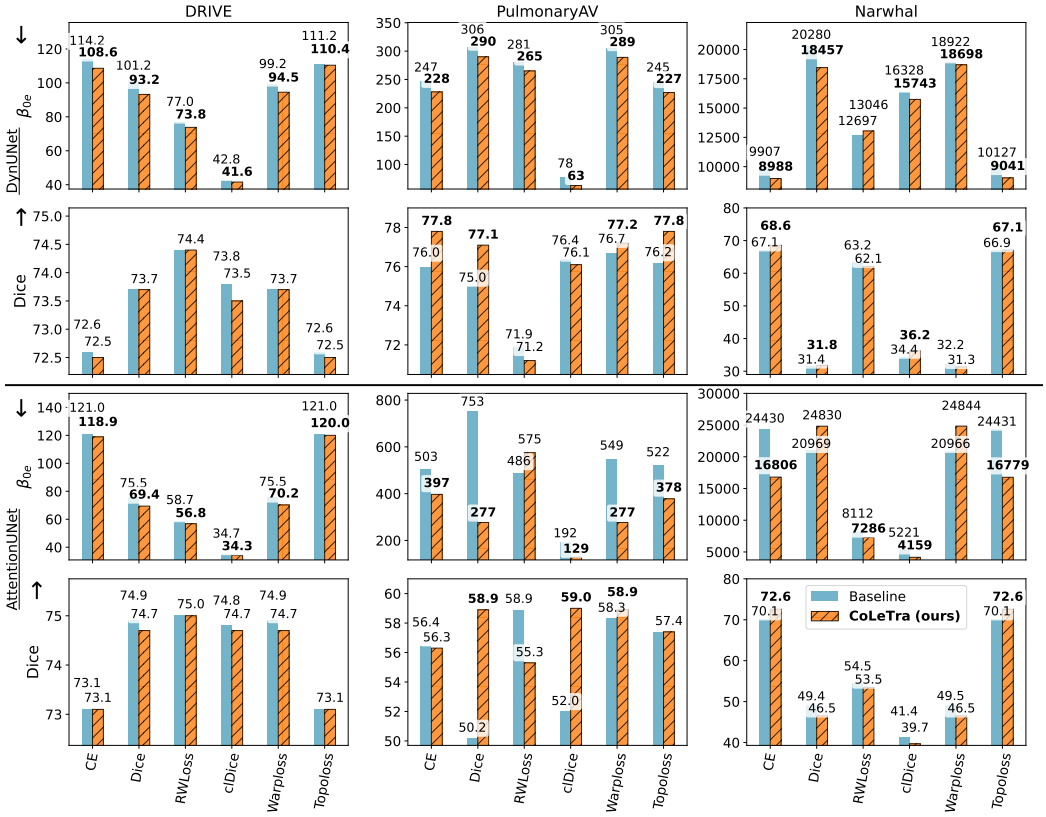

Figure 3: Comparison of baselines (three datasets, two architectures, six loss functions) with and without CoLeTra. Table with all performance metrics in Appendix E.

space limitations, here, we report the first global Betti error (β_{0e}) as it is the most relevant for all tasks, while the other Betti errors can be found in Appendix D.

CoLeTra generally led to **smaller Betti errors** across all datasets, loss functions, and architectures, indicating higher topology accuracy (see Figure 3, Figure 4, Appendix E). These improvements were observed even when optimizing topology loss functions: CoLeTra always improved Topoloss segmentations, and it improved clDice and Warping loss in every case except when optimizing AttentionUNet on the Narwhal dataset. When optimizing non-topology loss functions (CE, Dice, and RWLoss), CoLeTra consistently yielded topologically more accurate segmentations on DRIVE dataset, and in the majority of the cases in PulmonaryAV and the Narwhal dataset. Notably, the smallest Betti errors across all three datasets were achieved with CoLeTra. Moreover, CoLeTra did not appear to create false cycles since the second global and local Betti errors (β_{1e}) also generally decreased (Appendix D). Furthermore, in many cases, CoLeTra led to better clDice and Dice coefficients (Section E).

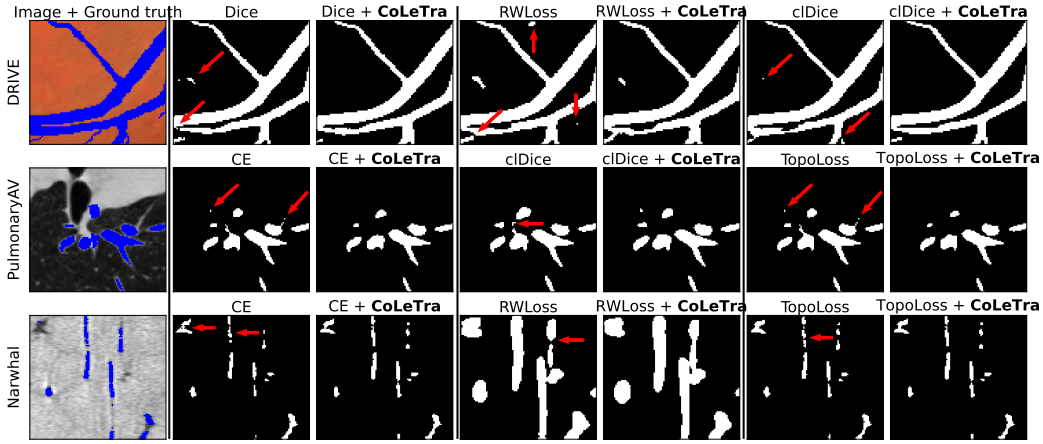

Figure 4: Example segmentations on four datasets and the best-performing loss functions, with and without CoLeTra. Arrows: topological errors fixed by CoLeTra.

4.2. Sensitivity analysis

We investigated how the two main hyper-parameters of CoLeTra—the size and number of patches—affect the topology accuracy gains. To this end, we optimized DynUNet on the DRIVE dataset with the Dice loss as described in Section 4.1. In addition to measuring the Betti error, we counted the number of small (< 30 pixels) connected components, as these structures represent fragments of broken blood vessels. With the DRIVE images containing, on average, only one or two small connected components per image according to the ground truth, a small number of these connected components in the segmentation masks indicates high topology accuracy. We explored the number of patches $n = \{10, 20, \ldots, 200\}$ and the patch sizes $s = \{3, 7, \ldots, 67\}$ in two separate sets of experiments.

CoLeTra decreased the Betti error and number of small connected components with patch sizes between 10 and 120 (see Figure 5 (a,b)). Beyond 120 patches, the Betti error and number of connected components oscillated. Similarly, the patch sizes between 3^2 and 19^2 decreased the Betti error and the number of small connected components (see Figure 5 (c,d). Thereafter, these metrics also started to oscillate. Figure 5 also shows that a moderately different patch size and number of patches from the ones we used in Section 4.1 ($n = 63, s = 15$) leads to similar topology accuracy.

5. Discussion

We introduced CoLeTra, a data augmentation strategy for improving topology accuracy by incorporating the prior knowledge that structures that appear disconnected can be connected. We evaluated CoLeTra on three benchmark datasets with different architectures and loss functions, and we studied its sensitivity to several hyper-parameter choices.

CoLeTra **consistently** improved topology accuracy across different architectures and loss functions on datasets with topologically accurate (DRIVE, PulmonaryAV) and inaccurate training labels (Narwhal). CoLeTra not only reduced the Betti errors but, in many

cases, it also improved clDice and Dice coefficient (see Section E). It is important to note, however, that while improvements in non-topology metrics are a beneficial side effect, they are not the primary focus of CoLeTra. Instead, CoLeTra emphasizes topology accuracy, which is critical for downstream quantification tasks, such as counting the number of connected components. In some experimental scenarios, the performance gains were moderate, which aligns with the expected outcome of adding just one extra data augmentation transformation on top of several other data augmentations during training. A single data augmentation technique alone typically does not lead to very large performance improvements (Nanni et al., 2021; Mumuni and Mumuni, 2022)—even advanced strategies, such as CutOut and MixUp (see Yun et al. (2019)).

CoLeTra demonstrates, for the first time, that topology can be improved via data augmentation. Although previous works have also used image inpainting to increase datasets' diversity (see Section 2), CoLeTra is unique in that it is specifically designed to improve topology accuracy and it demonstrates to do so, even when optimizing topology loss functions. CoLeTra's improvements demonstrate that data augmentation methods can complement topology loss functions and, potentially, deep learning architectures and postprocessing methods that focus on topology (*e.g.*, Jin et al. (2019); Yang et al. (2022); Qi et al. (2023); Carneiro-Esteves et al. (2024)). This capability to enhance topology accuracy alongside other methods during training is important since typical deep learning methods cannot guarantee accurate topology unless the topological features are known beforehand (Clough et al., 2020), which does not often occur. Moreover, CoLeTra was particularly useful when the dataset training labels were topologically incorrect; in the Narwhal dataset, CE and Topoloss achieved the best segmentations according to the standard metrics (Dice, clDice) and, after applying CoLeTra, topology accuracy (the Betti error) further improved.

CoLeTra's simplicity ensures that CoLeTra is fast, while requiring no extra GPU memory. This is crucial in scenarios where other topology-enhancing methods are limited by their computational cost (Hu et al., 2019; Hu, 2022; Liao, 2023) or larger GPU memory requirements (Shit et al., 2021). Additionally, CoLeTra's simple strategy to determine where to break the tubular structures can be extended to focus on fixing specific type of breaks (Ren et al., 2024) or in other locations considered important. Exploring performance gains in particular cases was beyond the scope of our study; instead, we demonstrated that even a simple approach can improve topology accuracy.

CoLeTra has two main hyper-parameters: the size, and the number of patches that will disconnect the tubular structures. In our comparisons across architectures, datasets, and loss functions, we utilized the hyper-parameter values that we found on a very small grid search—likely a sub-optimal configuration that, nevertheless, increased the topology accuracy of the segmentations. More accurate segmentations could be achieved with a more exhaustive grid search, and by jointly accounting for the datasets' particularities, such as the disproportion between the pixels from thin and thick structures (Liew et al., 2021), class imbalance, and noisy labels (Song et al., 2022).

Our sensitivity analysis showed that a large number of patches and a large patch size tend to decrease the Betti error and the number of small connected components, indicating higher topology accuracy. The range of patch sizes that improved topology accuracy (from 3^2 to 23^2, Figure 5 (c,d)) overlaps with the range of the widths of the blood vessels (from 2 to 18.5), demonstrating that patch size can be tuned, intuitively, based on the structures'

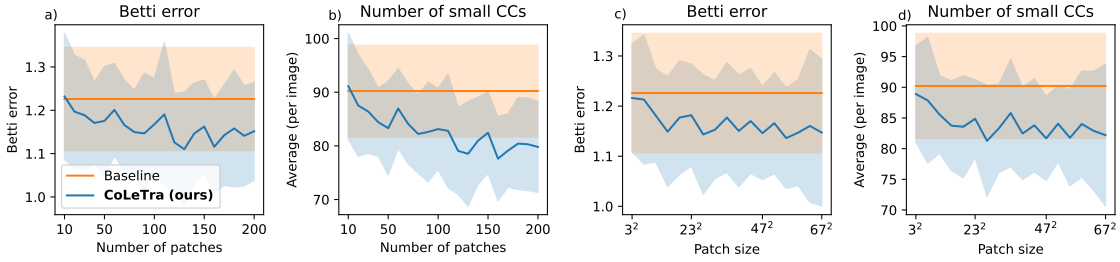

Figure 5: Betti error and number of small (< 30 pixels) connected components achieved with CoLeTra on the DRIVE dataset with different number of patches (a,b) and patch sizes (c,d). (CC: connected components)

width. Regarding the number of patches, we hypothesize that the optimal value depends on the accuracy of the annotations. In datasets with topologically inaccurate labels marking visually-connected structures as disconnected (see Figure 1 (Narwhal)), the optimization will reinforce such incorrect disconnection. Since CoLeTra promotes the opposite (*i.e.*, connecting visually-disconnected structures), a large number of patches may help compensating the structures mislabeled as disconnected.

6. Conclusion

CoLeTra increases topology accuracy in a wide range of settings, including different architectures, loss functions, and datasets with topologically accurate and inaccurate labels. Additionally, it does not deteriorate—often, it improves—other metrics, such as Dice coefficient and clDice. Our sensitivity analysis showed that CoLeTra is robust to different hyper-parameter choices, and these hyper-parameters are largely intuitive, facilitating their tuning. Finally, to further encourage research in the domain of image segmentation and topology, we release the Narwhal dataset.

Limitations The advantageousness of CoLeTra, as any other data augmentation, will depend on the dataset and task. CoLeTra's advantageous capability of promoting the connectivity between visually-disconnected structures may be disadvantageous on datasets with visually-disconnected structures that should remain disconnected. Additionally, CoLeTra assumes that false positives (*i.e.*, incorrectly connected structures) are relatively rare in the training labels, i.e., that the training labels indicate entire and/or broken structures.

Acknowledgments

This work was supported by Villum Foundation and NordForsk.

References

Nick Byrne, James R Clough, Israel Valverde, Giovanni Montana, and Andrew P King. A persistent homology-based topological loss for cnn-based multiclass segmentation of cmr. *IEEE transactions on medical imaging*, 42(1):3–14, 2022.

Sophie Carneiro-Esteves, Antoine Vacavant, and Odyssée Merveille. Restoring connectivity in vascular segmentation using a learned post-processing model. *arXiv preprint arXiv:2404.10506*, 2024.

Liang Chen, Paul Bentley, Kensaku Mori, Kazunari Misawa, Michitaka Fujiwara, and Daniel Rueckert. Self-supervised learning for medical image analysis using image context restoration. *Medical image analysis*, 58:101539, 2019.

Hongyu Cheng, Limin Zheng, Zeyu Yan, Haoran Zhang, Bo Meng, and Xiaowei Xu. Fusion of machine learning and deep neural networks for pulmonary arteries and veins segmentation in lung cancer surgery planning. In *International Conference on Pattern Recognition*, pages 422–438. Springer, 2024.

James R Clough, Nicholas Byrne, Ilkay Oksuz, Veronika A Zimmer, Julia A Schnabel, and Andrew P King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):8766–8778, 2020.

Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & computational geometry*, 28:511–533, 2002.

Rickard Brüel Gabrielsson, Bradley J Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1553–1563. PMLR, 2020.

Hongliang He, Jun Wang, Pengxu Wei, Fan Xu, Xiangyang Ji, Chang Liu, and Jie Chen. Toposeg: Topology-aware nuclear instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21307–21316, 2023.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

Songping He, Yi Zou, Bin Li, Fangyu Peng, Xia Lu, Hui Guo, Xin Tan, and Yanyan Chen. An image inpainting-based data augmentation method for improved sclerosed glomerular identification performance with the segmentation model efficientnetb3-unet. *Scientific Reports*, 14(1):1033, 2024.

Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. *Advances in neural information processing systems*, 30, 2017.

Xiaoling Hu. Structure-aware image segmentation with homotopy warping. *Advances in Neural Information Processing Systems*, 35:24046–24059, 2022.

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. *Advances in neural information processing systems*, 32, 2019.

Xiaoling Hu, Dimitris Samaras, and Chao Chen. Learning probabilistic topological representations using discrete morse theory. *arXiv preprint arXiv:2206.01742*, 2022.

Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

Umair Javaid, Damien Dasnoy, and John A Lee. Semantic segmentation of computed tomography for radiotherapy with deep learning: compensating insufficient annotation quality using contour augmentation. In *Medical Imaging 2019: Image Processing*, volume 10949, pages 682–694. SPIE, 2019.

Qiangguo Jin, Zhaopeng Meng, Tuan D Pham, Qi Chen, Leyi Wei, and Ran Su. Dunet: A deformable network for retinal vessel segmentation. *Knowledge-Based Systems*, 178: 149–162, 2019.

Yannick Kirchhoff, Maximilian R Rokuss, Saikat Roy, Balint Kovacs, Constantin Ulrich, Tassilo Wald, Maximilian Zenk, Philipp Vollmuth, Jens Kleesiek, Fabian Isensee, et al. Skeleton recall loss for connectivity conserving and resource efficient segmentation of thin tubular structures. In *European Conference on Computer Vision*, pages 218–234. Springer, 2024.

Junjie Li, Zilei Wang, and Xiaoming Hu. Learning intact features by erasing-inpainting for few-shot classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8401–8409, 2021.

Wei Liao. Segmentation of tubular structures using iterative training with tailored samples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23643–23652, 2023.

Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, and Jiashi Feng. Deep interactive thin object selection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 305–314, 2021.

Manxi Lin, Kilian Zepf, Anders Nymark Christensen, Zahra Bashir, Morten Bo Søndergaard Svendsen, Martin Tolsgaard, and Aasa Feragen. Dtu-net: learning topological similarity for curvilinear structure segmentation. In *International Conference on Information Processing in Medical Imaging*, pages 654–666. Springer, 2023.

Chuni Liu, Boyuan Ma, Xiaojuan Ban, Yujie Xie, Hao Wang, Weihua Xue, Jingchao Ma, and Ke Xu. Enhancing boundary segmentation for topological accuracy with skeleton-based methods. *arXiv preprint arXiv:2404.18539*, 2024.

Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The gudhi library: Simplicial complexes and persistent homology. In *Mathematical Software–ICMS 2014: 4th International Congress, Seoul, South Korea, August 5-9, 2014. Proceedings 4*, pages 167–174. Springer, 2014.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542, 2021.

Alhassan Mumuni and Fuseini Mumuni. Data augmentation: A comprehensive survey of modern approaches. *Array*, 16:100258, 2022.

Andriy Myronenko. 3d mri brain tumor segmentation using autoencoder regularization. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4*, pages 311–320. Springer, 2019.

Loris Nanni, Michelangelo Paci, Sheryl Brahnam, and Alessandra Lumini. Comparison of different image data augmentation approaches. *Journal of imaging*, 7(12):254, 2021.

Yunhao Nie, Bo Lu, Qiyuan Chen, Qinghai Miao, and Yisheng Lv. Synposes: Generating virtual dataset for pedestrian detection in corner cases. *IEEE Journal of Radio Frequency Identification*, 6:801–804, 2022.

Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

Doruk Oner, Adélie Garin, Mateusz Koziński, Kathryn Hess, and Pascal Fua. Persistent homology with improved locality information for more effective delineation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

Yaolei Qi, Yuting He, Xiaoming Qi, Yuan Zhang, and Guanyu Yang. Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6070–6079, 2023.

Jiaxiang Ren, Zhenghong Li, Wensheng Cheng, Zhilin Zou, Kicheon Park, Yingtian Pan, and Haibin Ling. Self-supervised 3d skeleton completion for vascular structures. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 579–589. Springer, 2024.

Daniel V Ruiz, Bruno A Krinski, and Eduardo Todt. Anda: A novel data augmentation technique applied to salient object detection. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 487–492. IEEE, 2019.

Daniel V Ruiz, Bruno A Krinski, and Eduardo Todt. Ida: Improved data augmentation applied to salient object detection. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 210–217. IEEE, 2020.

Plabon Kumar Saha, Sinthia Ahmed, Tajbiul Ahmed, Hasidul Islam, Al Imran, AZM Tahmidul Kabir, et al. Data augmentation technique to expand road dataset using mask rcnn and image inpainting. In *2021 International Conference on Intelligent Technologies (CONIT)*, pages 1–6. IEEE, 2021.

Jialin Shi, Kailai Zhang, Chenyi Guo, Youquan Yang, Yali Xu, and Ji Wu. A survey of label-noise deep learning for medical image analysis. *Medical Image Analysis*, 95:103166, 2024a.

Pengcheng Shi, Jiesi Hu, Yanwu Yang, Zilve Gao, Wei Liu, and Ting Ma. Centerline boundary dice loss for vascular segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 46–56. Springer, 2024b.

Seung Yeon Shin, Sungwon Lee, Daniel Elton, James L Gulley, and Ronald M Summers. Deep small bowel segmentation with cylindrical topological constraints. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23*, pages 207–215. Springer, 2020.

Suprosanna Shit, Johannes C Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien PW Pluim, Ulrich Bauer, and Bjoern H Menze. cldice-a novel topology-preserving loss function for tubular structure segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16560–16569, 2021.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 2022.

Joes Staal, Michael D Abràmoff, Meindert Niemeijer, Max A Viergever, and Bram Van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging*, 23(4):501–509, 2004.

Nico Stucki, Johannes C Paetzold, Suprosanna Shit, Bjoern Menze, and Ulrich Bauer. Topologically faithful image segmentation via induced matching of persistence barcodes. In *International Conference on Machine Learning*, pages 32698–32727. PMLR, 2023.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022.

Juan Miguel Valverde and Jussi Tohka. Region-wise loss for biomedical image segmentation. *Pattern Recognition*, 136:109208, 2023.

Mario Viti, Hugues Talbot, Bassam Abdallah, Etienne Perot, and Nicolas Gogin. Coronary artery centerline tracking with the morphological skeleton loss. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2741–2745. IEEE, 2022.

Shiyao Wang, Qi Liu, Yicheng Zhong, Zhilong Zhou, Tiezheng Ge, Defu Lian, and Yuning Jiang. Creagan: An automatic creative generation framework for display advertising. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 7261–7269, 2022.

Nina Kølln Wittig, Maja Østergaard, Jonas Palle, Thorbjørn Erik Køppen Christensen, Bente Lomholt Langdahl, Lars Rejnmark, Ellen-Margrethe Hauge, Annemarie Brüel, Jesper Skovhus Thomsen, and Henrik Birkedal. Opportunities for biomineralization research using multiscale computed x-ray tomography as exemplified by bone imaging. *Journal of Structural Biology*, 214(1):107822, 2022.

Meilong Xu, Xiaoling Hu, Saumya Gupta, Shahira Abousamra, and Chao Chen. Semi-supervised segmentation of histopathology images with noise-aware topological consistency. In *European Conference on Computer Vision*, pages 271–289. Springer, 2024.

Benyi Yang, Zhenyu Liu, Guifang Duan, and Jianrong Tan. Mask2defect: A prior knowledge-based data augmentation method for metal surface defect inspection. *IEEE Transactions on Industrial Informatics*, 18(10):6743–6755, 2021.

Xin Yang, Zhiqiang Li, Yingqing Guo, and Dake Zhou. Dcu-net: A deformable convolutional neural network based on cascade u-net for retinal vessel segmentation. *Multimedia Tools and Applications*, 81(11):15593–15607, 2022.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

I Zanette, B Enders, M Dierolf, P Thibault, Regine Gradl, Ana Diaz, Manuel Guizar-Sicairos, A Menzel, Franz Pfeiffer, and Paul Zaslansky. Ptychographic x-ray nanotomography quantifies mineral distributions in human dentine. *Scientific Reports*, 5(1):9210, 2015.

Lingzhi Zhang, Tarmily Wen, Jie Min, Jiancong Wang, David Han, and Jianbo Shi. Learning object placement by inpainting for compositional data augmentation. In *Computer*

Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16, pages 566–581. Springer, 2020.

Minghui Zhang, Yaoyu Liu, Junyang Wu, Xin You, Hanxiao Zhang, Junjun He, and Yun Gu. Toposculpt: Betti-steered topological sculpting of 3d fine-grained tubular shapes. *arXiv preprint arXiv:2509.03938*, 2025.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

Appendix A. The Narwhal dataset

A.1. Image acquisition and reconstruction

Narwhal tusks, composed of an inner dentine ring with pores (dentine tubules) running through it (Zanette et al., 2015), can be effectively studied using microcomputed tomography, particularly synchrotron radiation microcomputed tomography (SR-µCT) (Wittig et al., 2022). In order to study its internal structure, a piece of narwhal tusk was cut radially, with respect to the tusk length, to a square rod with side lengths 460 µm by 530 µm using a diamond blade saw (Accutom-5, Struers, Ballerup, Denmark). The rod was imaged with SR-µCT in the Paul Scherrer Institue, Viligen, Switzerland, at the TOMCAT beamline X02DA of the Swiss Light Source. Several scans were measured along the long axis of the rod to image it entirely. For each scan, 2000 projections were collected covering 180° with an energy of 18 keV, and an exposure time of 150 ms. 30 closed beam images were used for dark field correction and 50 open beam images for flat field correction. A 20 µm LuAG:Ce scintillator, a microscope (Optique Peter, France) with a ×20 magnification, and a PCO.Edge 5.5 camera (PCO AG, Kelheim, Germany) gave rise to an isotropic voxel size of 0.325 µm.

Image reconstruction was performed on-site using the GridRec algorithm in the ReconManagerRa pipeline plug-in for ImageJ. In the reconstruction, a Parzen filter with a cut-off frequency of 0.5 was employed as well as standard ring removal. The data was scaled to be between $-4 * 10^{-4}$ and $8 * 10^{-4}$, and output as 16-bit grey level images. Each scan had dimensions of 2560 × 2560 × 2160 voxels.

A.2. Training, validation, and test splits

For the purpose of this study, we employed five scans containing abundant dentine tubules. One scan, used for training, was divided into 64 overlapping patches. Then, we discarded those patches containing background area, leading to our training set of 20 patches of, approximately, 591 × 530 × 583 voxels. The validation set corresponds to a 500 × 500 × 500 patch from another scan, and the test set is comprised by three 500 × 500 × 500 patches, each from a different scan.

A.3. Labels

Due to the extremely large size of the images, making their manual annotation prohibitively expensive, we derived pseudo-labels for the training set and pseudo-ground truths for the validation and test set. The pseudo-labels were obtained via thresholding to simulate the common practice employed by researchers when analyzing these type of images. The pseudo-ground truths, used exclusively to evaluate the predictions, were derived through a more elaborated process yielding segmentation masks that, although they were not perfect, they permit measuring topology accuracy reliably (i.e., counting the number of tubules), and permit quantifying tubules' properties, such as their directionality.

Pseudo-labels (training set) First, we thresholded the images and considered as "dentine tubule" those voxels with intensities below 17,450—threshold that we found experimentally. Then, we removed non-tubular areas that we identified with a distance transform.

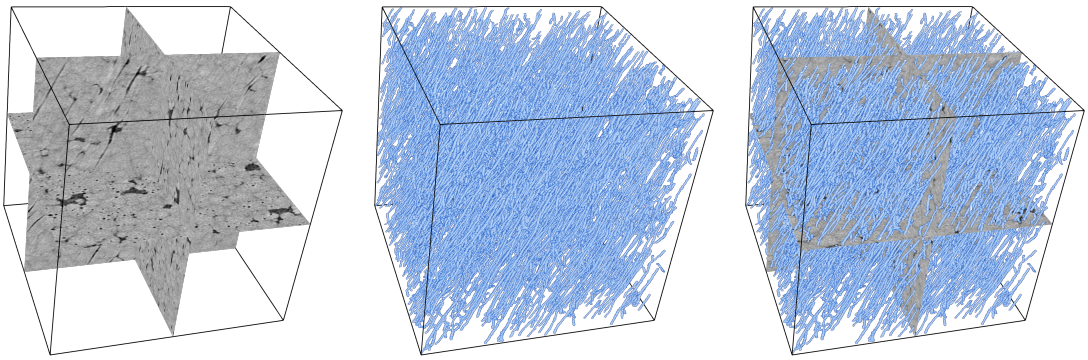

Figure 6: Example of labels in the Narwhal dataset. Left shows orthogonal central slices from the volume, middle shows the labels, and right shows the two together.

Pseudo-ground truth (validation and test set) The pseudo-ground truth labels are created by first tracking the tubules through the volume and then assigning labels to the tracks. In unmineralized regions of the tusk, the intensity of the tubule and the regions are the same, which makes segmentation challenging. By utilizing that the dentine tubules are running predominantly along the z-direction of the volume, we can track the tubules to obtain their center lines.

Tracking is done by initially detecting the tubule's center position and connecting detected points moving along the z-direction. The center position is detected as the local maxima in a Gaussian-smoothed image ($\sigma = 1.5$). We connect mutually closest points in consecutive frames under some constraints. Points in one frame must be separated by an Euclidean distance of at least five voxels, be within an Euclidean distance of no more than three voxels in consecutive frames, if no points are detected within five frames tracks are stopped, and we only keep tracks that are at least 20 voxels long (Euclidean distance). Based on the tracks, we create a label image containing a thresholded version of the original image (threshold value of 30,720) but masked along the tracks. We employ a minimum mask of a 3×3 window that the label should minimally be, and a maximum mask of a circle with a diameter of 5×5 that the label can maximally obtain. If the thresholded image is within these limits, the threshold will be chosen. After labeling, we see tubules being tracked nicely through the unmineralized regions, and labels that connect in a complex network as you will expect for dentine tubules. Tracks are illustrated in Figure 6.

A.4. Dataset availability

The Narwhal dataset, released under CC BY-NC 4.0 licence, can be downloaded at https://archive.compute.dtu.dk/files/public/projects/CoLeTra_Narwhal.

Appendix B. Optimization

B.1. Architectures

We optimized two model architectures: DynUNet[1] and AttentionUNet[2].

DynUNet had the same architecture in all our experiments, differing only in the dimension of the kernels and the input channels. In DRIVE dataset we used 2D kernels ($spatial\_dims = 2$), and in PulmonaryAV and the Narwhal dataset we used 3D kernels ($spatial\_dims = 3$). DRIVE contained RGB images ($in\_channels = 3$) whereas PulmonaryAV and the Narwhal dataset contained single-channel images ($in\_channels = 1$). DynUNet had four downsampling blocks that used convolutions with strides = 2 to reduce the dimensionality of the features. Additionally, DynUNet employed deep supervision, optimizing the output of three upsampling blocks. The exact configuration of DynUNet was:

```
DynUNet(spatial_dims, in_channels,
    out_channels=2,
    kernel_size=[3, 3, 3, 3, 3, 3],
    strides=[1, 2, 2, 2, 2, 2],
    upsample_kernel_size=[2, 2, 2, 2, 2],
    deep_supervision=True,
    deep_supr_num=3)
```

AttentionUNet architecture differed across datasets. We initially aimed to configure AttentionUNet to match, as much as possible, the number of parameters of DynUNet. However, due to the larger GPU memory requirements, we had to use a smaller and shallower architecture in some datasets. The dimension of the kernels and the input channels were as in DynUNet. Additionally, the depth and the number of feature maps per block were as follows:

DRIVE

```
AttentionUNet(spatial_dims=2,
    in_channels=3, out_channels=2,
    channels=[32, 64, 128, 256, 512, 512],
    strides=[1, 2, 2, 2, 2, 2])
```

PulmonaryAV and Narwhal

```
AttentionUNet(spatial_dims=3,
    in_channels=1, out_channels=2,
    channels=[16, 24, 32],
    strides=[1, 2, 2])
```

B.2. Data augmentation and sampling strategy

All images were randomly augmented, with a 50% probability in each transform independently, with random Gaussian noise ($\mu = 0, \sigma = 1$), random gamma correction ($\gamma \in$

1. https://docs.monai.io/en/stable/networks.html#dynunet
2. https://docs.monai.io/en/stable/networks.html#attentionunet

Table 1: DRIVE

s	n	✓?
11	94	
11	117	
11	141	
15	50	
15	63	D
15	75	
19	31	
19	39	A
19	47	

Table 2: PulmonaryAV

s	n	✓?
11	10	
11	13	
11	15	
15	4	
15	5	
15	6	A
19	2	D
19	3	

$[0.8, 1.2]$), and random axis flip. DRIVE images were, in addition, randomly re-scaled between $[1, 1.75]$. Since PulmonaryAV and Narwhal volumes were too big to fit the GPU memory, we sampled, after applying the Gaussian noise, 15 $128 \times 128 \times 128$ patches in both datasets. We sampled only 15 patches to ensure diversity in the data augmentation; note that we optimized the models for a certain number of iterations and not epochs.

B.3. Optimization

We utilized Adam optimizer with a starting learning rate of 0.001 and weight decay of 10^{-5}. We decreased the learning rate during training: $\eta \leftarrow \eta(1 - \frac{iteration}{max\_iterations})^{0.9}$. For DRIVE, PulmonaryAV, and the Narwhal dataset, we optimized the models for 5000, 7000, 1000 iterations, respectively. In addition, due to the class imbalance in the Narwhal dataset, we used weighted Cross entropy (including in Topoloss that relies on Cross entropy) with the following weights: $[0.05, 0.95]$. The rectified region wise maps employed by RWLoss in the Narwhal dataset were also multiplied by those weights accordingly.

Appendix C. CoLeTra's hyper-parameters search

For each dataset and architecture, we explored a maximum of nine different configurations combining three patch sizes (s) and three number of patches (n). For the patch sizes, we explored the values 11, 15, and 19 since the structures to be segmented have a width around that range. For the number of patches, to avoid disconnecting too many or too few structures, we designed a simple strategy based on the number of pixels in the annotated areas. Given the average number of pixels and a patch size, consider the number of patches such that the maximum covered area of the structures after applying CoLeTra corresponds to approximately 40%, 50%, and 60% of the total.

Tables 1, 2 and 3 specify the combinations of hyper-parameters that were explored, and which one yielded the best performance in DynUNet (D) and AttentionUNet (A) architectures. Note that, in the Narwhal dataset, a single patch size of $15 \times 15 \times 15$ and $19 \times 19 \times 19$ already covers more than the expected number voxels from the structure to be found in the $128 \times 128 \times 128$ patches used during training. Thus, we only explored to apply a single patch when its size was 15 or 19.

Table 3: Narwhal

s	n	✓?
11	2	D
11	3	
11	4	
15	1	A
19	1	

Appendix D. Betti Errors

Tables 4 and 5 show all the Betti errors (global and local) obtained in our experiments, which were computed via the Gudhi library (Maria et al., 2014). The global Betti errors were the difference in the Betti numbers between the entire ground-truth annotation and the prediction. The local Betti errors were computed similarly but in patches of size 64×64 patches (2D images) or $48 \times 48 \times 48$ (3D images). The patches were extracted either randomly or in a sliding-window manner depending on whether the total number of non-overlapping patches that could be extracted from the images were more than 500 (for 2D images) or 100 (for 3D images).

Table 4: Global and local Betti Errors (DynUNet)

			β_{0e} (global)	β_{1e} (global)	β_{2e} (global)	β_{0e} (local)	β_{1e} (local)	β_{2e} (local)
DRIVE	CE	baseline	114.2 (11.8)	35.5 (2.4)	—	1.3 (0.1)	0.2 (0.0)	—
		+**CoLeTra**	**108.6** (10.2)	**33.6** (2.9)	—	**1.3** (0.1)	0.2 (0.0)	—
	Dice	baseline	101.2 (10.8)	28.5 (3.3)	—	1.2 (0.1)	0.2 (0.0)	—
		+**CoLeTra**	**93.2** (9.2)	**26.2** (3.6)	—	**1.1** (0.1)	0.2 (0.0)	—
	RWLoss	baseline	77.0 (8.2)	29.1 (2.8)	—	1.0 (0.1)	0.2 (0.0)	—
		+**CoLeTra**	**73.8** (6.2)	**28.2** (3.1)	—	**1.0** (0.1)	0.2 (0.0)	—
	clDice	baseline	42.8 (7.3)	24.0 (3.2)	—	0.7 (0.1)	0.2 (0.0)	—
		+**CoLeTra**	**41.6** (6.6)	**22.1** (3.4)	—	**0.7** (0.1)	0.2 (0.0)	—
	Warploss	baseline	99.2 (9.2)	28.0 (3.8)	—	1.2 (0.1)	0.2 (0.0)	—
		+**CoLeTra**	**94.5** (10.0)	**25.7** (2.8)	—	**1.2** (0.1)	0.2 (0.0)	—
	Topoloss	baseline	111.2 (13.4)	35.0 (2.3)	—	1.3 (0.1)	0.2 (0.0)	—
		+**CoLeTra**	**110.4** (11.3)	**33.6** (2.8)	—	**1.3** (0.1)	0.2 (0.0)	—
PulmonaryAV	CE	baseline	247 (14)	26 (5)	14 (0)	17 (0)	4 (0)	3 (0)
		+**CoLeTra**	**228** (15)	27 (4)	15 (0)	**15** (1)	4 (0)	3 (0)
	Dice	baseline	306 (27)	24 (5)	13 (1)	18 (1)	4 (0)	3 (0)
		+**CoLeTra**	**290** (25)	25 (2)	14 (0)	**17** (0)	4 (0)	3 (0)
	RWLoss	baseline	281 (26)	25 (4)	14 (1)	17 (1)	4 (0)	3 (0)
		+**CoLeTra**	**265** (21)	26 (5)	15 (0)	**17** (0)	**4** (0)	3 (0)
	clDice	baseline	78 (12)	24 (4)	14 (1)	5 (0)	4 (0)	3 (0)
		+**CoLeTra**	**63** (6)	25 (2)	14 (0)	**5** (0)	4 (0)	**3** (0)
	Warploss	baseline	305 (31)	24 (4)	14 (1)	18 (1)	4 (0)	3 (0)
		+**CoLeTra**	**289** (26)	24 (3)	14 (0)	**17** (1)	4 (0)	3 (0)
	Topoloss	baseline	245 (13)	25 (4)	14 (1)	16 (0)	4 (0)	3 (0)
		+**CoLeTra**	**227** (15)	27 (4)	14 (0)	**15** (1)	4 (0)	3 (0)
Narwhal	CE	baseline	9907 (1128)	336 (367)	45 (16)	56 (6)	5 (0)	0 (0)
		+**CoLeTra**	**8988** (429)	665 (288)	**36** (14)	**50** (3)	5 (1)	**0** (0)
	Dice	baseline	20280 (249)	1105 (25)	51 (0)	112 (3)	7 (0)	0 (0)
		+**CoLeTra**	**18457** (1220)	**1103** (33)	51 (0)	**100** (7)	**7** (0)	0 (0)
	RWLoss	baseline	12697 (2712)	806 (30)	51 (0)	73 (17)	5 (0)	0 (0)
		+**CoLeTra**	13046 (3102)	**707** (206)	**51** (0)	76 (19)	**5** (0)	0 (0)
	clDice	baseline	16328 (1693)	1146 (6)	51 (0)	87 (9)	7 (0)	0 (0)
		+**CoLeTra**	**15743** (592)	**1139** (12)	51 (0)	**84** (2)	**7** (0)	0 (0)
	Warploss	baseline	18922 (1388)	1103 (25)	51 (0)	103 (9)	7 (0)	0 (0)
		+**CoLeTra**	**18698** (781)	**1103** (35)	51 (0)	**102** (4)	**7** (0)	0 (0)
	Topoloss	baseline	10127 (1244)	416 (240)	70 (45)	57 (8)	5 (1)	0 (0)
		+**CoLeTra**	**9041** (682)	**414** (232)	**33** (23)	**50** (6)	**5** (1)	**0** (0)

Table 5: Global and local Betti Errors (AttentionUNet)

			β_{0e} (global)	β_{1e} (global)	β_{2e} (global)	β_{0e} (local)	β_{1e} (local)	β_{2e} (local)
DRIVE	CE	baseline	121.0 (13.6)	34.9 (3.1)	—	1.4 (0.1)	0.2 (0.0)	—
		+**CoLeTra**	**118.9** (13.7)	**32.2** (2.8)	—	**1.4** (0.2)	0.2 (0.0)	—
	Dice	baseline	75.5 (19.4)	29.2 (3.4)	—	1.0 (0.2)	0.2 (0.0)	—
		+**CoLeTra**	**69.4** (16.7)	**27.8** (3.4)	—	**0.9** (0.2)	0.2 (0.0)	—
	RWLoss	baseline	58.7 (11.7)	29.9 (3.1)	—	0.8 (0.1)	0.2 (0.0)	—
		+**CoLeTra**	**56.8** (11.2)	29.9 (3.0)	—	**0.8** (0.1)	0.2 (0.0)	—
	clDice	baseline	34.7 (5.2)	19.6 (4.9)	—	0.7 (0.1)	0.3 (0.0)	—
		+**CoLeTra**	**34.3** (5.4)	**19.1** (3.3)	—	0.7 (0.1)	**0.2** (0.0)	—
	Warploss	baseline	75.5 (19.4)	29.2 (3.4)	—	1.0 (0.2)	0.2 (0.0)	—
		+**CoLeTra**	**70.2** (16.6)	**28.0** (3.0)	—	**0.9** (0.2)	**0.2** (0.0)	—
	Topoloss	baseline	121.0 (13.6)	34.9 (3.1)	—	1.4 (0.1)	0.2 (0.0)	—
		+**CoLeTra**	**120.0** (13.7)	**32.7** (2.5)	—	**1.4** (0.2)	**0.2** (0.0)	—
PulmonaryAV	CE	baseline	503 (54)	159 (79)	22 (15)	23 (1)	5 (2)	2 (0)
		+**CoLeTra**	**397** (74)	**68** (37)	**15** (10)	**21** (4)	**4** (1)	3 (0)
	Dice	baseline	753 (309)	345 (279)	83 (98)	26 (5)	8 (5)	3 (2)
		+**CoLeTra**	**277** (107)	**33** (21)	**11** (3)	**17** (3)	**5** (0)	**3** (0)
	RWLoss	baseline	486 (215)	108 (116)	17 (16)	18 (5)	4 (1)	2 (0)
		+**CoLeTra**	575 (371)	**100** (106)	**13** (6)	20 (4)	5 (1)	3 (0)
	clDice	baseline	192 (59)	29 (20)	14 (6)	8 (1)	3 (0)	3 (0)
		+**CoLeTra**	**129** (31)	34 (7)	**14** (1)	9 (1)	5 (0)	3 (0)
	Warploss	baseline	549 (126)	125 (85)	17 (12)	22 (4)	5 (1)	2 (0)
		+**CoLeTra**	**277** (107)	**33** (21)	**11** (3)	**17** (3)	5 (0)	3 (0)
	Topoloss	baseline	522 (50)	187 (49)	25 (12)	23 (2)	6 (1)	2 (0)
		+**CoLeTra**	**378** (87)	**57** (39)	**11** (6)	**22** (3)	**3** (1)	3 (0)
Narwhal	CE	baseline	24430 (4492)	888 (450)	58 (39)	151 (32)	7 (1)	0 (0)
		+**CoLeTra**	**16806** (4805)	**810** (396)	255 (359)	**102** (33)	**6** (1)	1 (2)
	Dice	baseline	20969 (4201)	736 (511)	44 (17)	124 (33)	5 (1)	0 (0)
		+**CoLeTra**	24830 (10180)	765 (495)	191 (241)	146 (69)	5 (1)	1 (1)
	RWLoss	baseline	8112 (4885)	718 (448)	36 (13)	44 (30)	6 (1)	0 (0)
		+**CoLeTra**	**7286** (5773)	**673** (466)	**34** (14)	**43** (28)	**5** (1)	0 (0)
	clDice	baseline	5221 (3044)	620 (316)	37 (9)	22 (17)	6 (0)	0 (0)
		+**CoLeTra**	**4159** (965)	**610** (335)	39 (11)	**16** (4)	6 (0)	0 (0)
	Warploss	baseline	20966 (4222)	735 (511)	44 (17)	124 (33)	5 (1)	0 (0)
		+**CoLeTra**	24844 (10194)	765 (494)	190 (240)	146 (69)	5 (1)	1 (1)
	Topoloss	baseline	24431 (4494)	888 (451)	57 (39)	151 (32)	**7** (1)	0 (0)
		+**CoLeTra**	**16779** (4860)	**822** (393)	267 (379)	**102** (33)	**6** (1)	1 (2)

Appendix E. Performance metrics

Table 6 shows the global Betti 0 error (β_{0e}), Dice coefficient and clDice of our baseline models trained with extensive data augmentation and when they, additionally, used our CoLeTra data augmentation strategy.

Table 6: Performance comparison between baseline models (trained with extensive data augmentation) and our CoLeTra on the two architectures, three datasets, and six loss functions.

			DynUNet			AttentionUNet		
			β_{0e}	Dice	clDice	β_{0e}	Dice	clDice
DRIVE	CE	baseline	114.2 (11.8)	72.6 (0.4)	70.0 (0.7)	121.0 (13.6)	73.1 (0.5)	70.7 (0.7)
		+**CoLeTra**	**108.6** (10.2)	72.5 (0.5)	**70.1** (0.6)	**118.9** (13.7)	73.1 (0.5)	**70.9** (0.7)
	Dice	baseline	101.2 (10.8)	73.7 (0.4)	71.8 (0.5)	75.5 (19.4)	74.9 (0.6)	73.4 (0.9)
		+**CoLeTra**	**93.2** (9.2)	73.7 (0.5)	**72.0** (0.6)	**69.4** (16.7)	74.7 (0.5)	73.4 (0.6)
	RWLoss	baseline	77.0 (8.2)	74.4 (0.4)	73.0 (0.6)	58.7 (11.7)	75.0 (0.3)	73.8 (0.4)
		+**CoLeTra**	**73.8** (6.2)	74.4 (0.3)	72.9 (0.5)	**56.8** (11.2)	75.0 (0.5)	73.8 (0.7)
	clDice	baseline	42.8 (7.3)	73.8 (0.3)	74.5 (0.4)	34.7 (5.2)	74.8 (0.3)	76.5 (0.4)
		+**CoLeTra**	**41.6** (6.6)	73.5 (0.3)	**74.6** (0.4)	**34.3** (5.4)	74.7 (0.3)	76.4 (0.4)
	Warploss	baseline	99.2 (9.2)	73.7 (0.4)	71.8 (0.5)	75.5 (19.4)	74.9 (0.6)	73.4 (0.9)
		+**CoLeTra**	**94.5** (10.0)	73.7 (0.5)	**72.0** (0.6)	**70.2** (16.6)	74.7 (0.4)	73.4 (0.5)
	Topoloss	baseline	111.2 (13.4)	72.6 (0.5)	70.0 (0.8)	121.0 (13.6)	73.1 (0.5)	70.7 (0.7)
		+**CoLeTra**	**110.4** (11.3)	72.5 (0.5)	**70.1** (0.6)	**120.0** (13.7)	73.1 (0.5)	**70.8** (0.7)
PulmonaryAV	CE	baseline	247 (14)	76.0 (1.5)	82.7 (1.1)	503 (54)	56.4 (2.2)	65.0 (3.5)
		+**CoLeTra**	**228** (15)	**77.8** (1.8)	**83.8** (1.2)	**397** (74)	56.3 (3.7)	**68.7** (3.1)
	Dice	baseline	306 (27)	75.0 (2.7)	81.0 (2.1)	753 (309)	50.2 (11.3)	55.1 (12.7)
		+**CoLeTra**	**290** (25)	**77.1** (1.8)	**82.7** (1.0)	**277** (107)	**58.9** (3.7)	**73.9** (3.8)
	RWLoss	baseline	281 (26)	71.9 (3.4)	81.8 (1.9)	486 (215)	58.9 (5.4)	66.2 (9.5)
		+**CoLeTra**	**265** (21)	71.2 (2.8)	81.6 (1.0)	575 (371)	55.3 (4.0)	**68.2** (6.2)
	clDice	baseline	78 (12)	76.4 (2.4)	85.1 (1.9)	192 (59)	52.0 (7.3)	76.4 (4.9)
		+**CoLeTra**	**63** (6)	76.1 (2.3)	**85.7** (1.5)	**129** (31)	**59.0** (2.0)	**80.2** (1.5)
	Warploss	baseline	305 (31)	76.7 (2.8)	81.7 (2.0)	549 (126)	58.3 (4.7)	66.5 (6.9)
		+**CoLeTra**	**289** (26)	**77.2** (1.9)	**82.8** (1.1)	**277** (107)	**58.9** (3.7)	**73.9** (3.8)
	Topoloss	baseline	245 (13)	76.2 (1.5)	82.7 (1.2)	522 (50)	57.4 (2.5)	63.8 (2.6)
		+**CoLeTra**	**227** (15)	**77.8** (1.8)	**83.8** (1.2)	**378** (87)	57.4 (2.6)	**69.9** (3.9)
Narwhal	CE	baseline	9907 (1128)	67.1 (5.0)	85.5 (1.3)	24430 (4492)	70.1 (3.1)	83.4 (2.0)
		+**CoLeTra**	**8988** (429)	**68.6** (4.4)	85.1 (1.4)	**16806** (4805)	**72.6** (1.0)	**85.1** (0.7)
	Dice	baseline	20280 (249)	31.4 (0.4)	67.0 (1.9)	20969 (4201)	49.4 (16.7)	77.5 (6.6)
		+**CoLeTra**	**18457** (1220)	**31.8** (2.0)	66.3 (2.8)	24830 (10180)	46.5 (14.6)	73.8 (8.7)
	RWLoss	baseline	12697 (2712)	63.2 (2.0)	83.3 (0.3)	8112 (4885)	54.5 (18.2)	83.7 (4.0)
		+**CoLeTra**	13046 (3102)	62.1 (0.5)	82.9 (0.2)	**7286** (5773)	53.5 (19.1)	83.7 (3.1)
	clDice	baseline	16328 (1693)	34.4 (4.0)	66.1 (4.4)	5221 (3044)	41.4 (9.4)	78.7 (4.4)
		+**CoLeTra**	**15743** (592)	**36.2** (2.1)	**67.4** (2.5)	**4159** (965)	39.7 (5.5)	**80.4** (4.1)
	Warploss	baseline	18922 (1388)	32.2 (1.5)	67.5 (1.8)	20966 (4222)	49.5 (16.7)	77.5 (6.6)
		+**CoLeTra**	**18698** (781)	31.3 (2.5)	66.2 (2.9)	24844 (10194)	46.5 (14.6)	73.8 (8.7)
	Topoloss	baseline	10127 (1244)	66.9 (5.3)	84.8 (2.2)	24431 (4494)	70.1 (3.1)	83.4 (2.0)
		+**CoLeTra**	**9041** (682)	**67.1** (4.6)	84.6 (1.0)	**16779** (4860)	**72.6** (1.0)	**85.1** (0.7)