A. APPENDIX

A.1. Critical pixel mask

Alg. 1 displays the pseudo-code for the extraction of our critical pixel mask.

Algorithm 1 CLoss critical pixel mask $\boldsymbol{M}$

Input 1: X % per-pixel logits of the prediction
Input 2: Y % ground truth mask
$X_{bin} \leftarrow argmax(X)$ % Binarize to foreground
$X_{fore} \leftarrow X(Foreground)$ % Foreground likelyhood
$S_X \leftarrow skeletonize(X_{fore})$ % Soft skeleton
$S_Y \leftarrow skeletonize(Y)$

Determine gaps % with distance transform (dTr)
$S_{gap} \leftarrow \max(0, S_Y - X_{bin})$
$V_{gap} \leftarrow \Big(\mathrm{dTr}\big(1 - (S_Y - S_{gap})\big) > \mathrm{dTr}\big(1 - S_{gap}\big)\Big) \odot Y$
% with Hadamard product $\odot$

Determine false positive connections
$S_{fp} \leftarrow \max(0, S_X - Y)$
$V_{fp} \leftarrow \Big(\mathrm{dTr}\big(1 - (S_X - S_{fp})\big) > \mathrm{dTr}\big(1 - S_{fp}\big)\Big) \odot X_{bin}$

Output: $\boldsymbol{M} \leftarrow V_{gap} \cup V_{fp}$

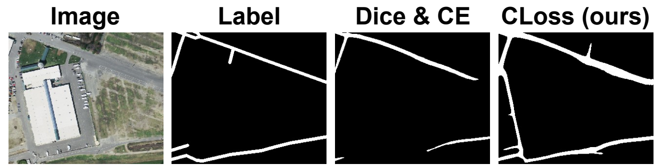
Applying soft skeletonization increases performance for binary segmentation over skeletonization of binary predictions [3–5]. Our proposed method can be implemented with any improved skeleton version of [3], eg. [4]. Skeletonization makes the extraction of the critical pixel mask orders of magnitudes faster than the recently proposed critical pixel mask extraction based on persistent homology [6], as compared in [13].

A.2. Metric Sensitivity to Gaps

Empirically, missed connections (gaps) in the predictions are far more frequent than false positive connections in our used datasets, see Fig. 5. Our proposed metrics help practitioners to identify methods that achieve superior gap closing, see Fig. 6. For completeness, we also emphasize the robustness of e_0-Gt to small separate false-positive predictions, which are a common issue for the e_0 metric [12].

A.3. Metric Susceptibility to Artifacts

It is of critical importance to use evaluation metrics with a high sensitivity towards the desired topological solution [11, 12]. However, popular topology metrics based on the number of connected components (β_0) are overly sensitive to artifacts, especially in the form of mico-noise (separate



Method	e_0 ↓	e_0-*Gt* ↓	clDice ↑	*AGS* ↑
Dice & CE	**0**	**0**	63.87	54.90
CLoss	2	**0**	**70.19**	**90.20**

Fig. 6: **Qualitative example of our proposed metrics.** Metric values are given for the displayed images. The metric values are colored for low (red&yellow) and high (green) sensitivity for the result with better gap closing (CLoss). Our proposed metrics e_0-Gt and AGS help to identify the subjectively preferred CLoss result. This example is taken from the Roads dataset images in Fig. 5.

components consisting of only a few pixels), which diminishes result integrity. The micro-noise problem is so far only addressed by [5], which removes small separate structures during training. Most recent solutions [3, 4, 6–8, 13, 14] don't address micro noise and are thereby effectively optimized for topological correctness and micro-noise suppression simultaneously. This is a problem because we are primarily interested in developing methods to achieve major topological correctness and not to achieve micro-noise suppression. We argue that micro-noise suppression should only be seen as a desired property of a method but not as a primary attribute to compare different methods for major topological correctness. We propose the following solution.

Proposed Topological Post-Processing. We propose to leverage the natural characteristic of overlap-based losses to minimize artifacts (and micro-noise) [1, 2]. Therefore, we propose to include a pretraining with a topology-insensitive overlap-based loss function by default and then conduct a fine-tuning with a topology-aware loss function. The suggested pretraining is compatible with all existing topology-preserving loss functions. We suggest to only keep the predictions of the fine-tuning training (X^{ft}_{bin}), which are connected to the pretraining predictions (X^{pre}_{bin}). Expressed as an equation, this means for the mask c_i of each connected component i in X^{ft}_{bin} we suggest to only keep i in X^{ft}_{bin} if

$$sum(c_i \odot X^{pre}_{bin}) > 0 \tag{5}$$

In particular, our proposed topological post-processing doesn't alter the connectivity of X^{opt}_{bin} in any way except for removing separate structures which were not present in the pretraining. Our suggested post-processing supports the intuition of improving the topology of an existing structure without adding new structures.

We note that a comparable effect of micro noise removal can in theory be achieved with a post-processing of opening and erosion operations. However, the applicability of opening and erosion operations depends highly on the predicted structures. Opening and erosion operations have a high risk of introducing new false positive connections that weren't present before the post-processing. Our suggested post-processing is by design robust to introducing new artifacts in the form of false positive connections.

Compared to the runtime solution of [5], our approach doesn't require any additional hyperparameters to select the size of the removed structures and will also remove artifacts which are missed by a fixed threshold.

A.4. Cement line dataset (CLD)

Image Acquisition. Specific steps for acquiring the used 3D images of the nanotomography bone cement line dataset are described in [18], sections 2.4.2 to 2.4.4.1. These steps include the preparation of the imaged specimen, the imaging procedure and the processing of the raw images. We cropped the images from their initial shape $1024^2 \times 1024$ to their final shape $1024^2 \times 600$ to focus on areas with pronounced cement lines and minimal artifacts.

Annotation. The annotation was done for each image on the complete initial shape of $1024^2 \times 1024$. The annotation of the cement lines was done manually in Avizo (version 2024.2, Thermo Fisher Scientific, Waltham, USA), utilizing the standard brush and interpolation tools. The cement lines have low contrast and diffuse borders. Importantly, their visibility and area proportion in the 2D view of the 3D image critically depends on the view angle in 3D, which doesn't always align well with the available view axes (xy,xz,zy). When the cement lines slightly change orientation inside the bone they can become much less pronounced in the current 2D view direction and significantly change in the displayed size.

To address these issues, we conducted a pre-annotation for all 3D images from all three view angles (axes:xy,xz,zy) to acquire a coarse initial representation of all cement lines. The pre-annotation corresponds to a detailed annotation about every 50 slices for each view axis for every 3D image. The pre-annotation was used to determine the preferred orientation of the majority of the cement lines within the 3D image. The pre-annotation was further used as an aid in the more thorough annotation to not miss any cement lines due to viewing angle effects. The thorough annotation was done along the most suitable view axis (xy,xz, or zy). This most suitable view axis was chosen in a way that the majority of the cement lines had a small area proportion in the 2D view of the 3D image, see Fig. 7. The thorough annotation corresponds to a detailed annotation about every three to five slices along the most suitable view axis for every 3D image. The thorough annotation for each 3D image was then interpolated with the standard Avizo interpolation tool. This interpolation was designed by Avizo to only interpolate between annotation slices without considering any gray values. However, the resulting annotation was a suitable compromise between annotation time and annotation accuracy.

Sometimes, the interpolation tool didn't work as expected, and there were holes within the annotation that didn't get interpolated. To close these interpolation holes, we conducted a series of morphological opening and closing operations until the holes were closed. We repeated this step with additional annotated slices when this approach didn't work initially. We note that the interpolation had also severe trouble interpolating more than one cement line at a time. As a solution, we conducted the thorough annotation and interpolation procedure only for selected parts of a single cement line at once, which we later combined into the complete annotation.

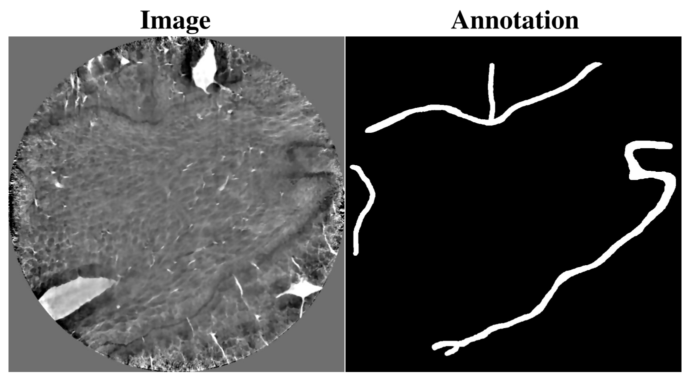


Fig. 7: **Cement line Annotation.** The cement lines have low contrast and diffuse borders. Also, their appearance is diffuse, note in the left image the changing grayscale characteristics of the cement line in the lower right corner from bottom to right.

A.5. Datasets

Roads contains aerial images (1500^2) of the road system in Massachusetts, similar to settings in [3] and [7] we use 421 images for training and 49 for testing, after excluding images with a white-masked pixel percentage of more than 4%. HRF-Retina consists of high-resolution fundus images (3504×2336) for retinal vessel segmentation, we use 36 images for training and 9 for testing. Vessap contains volumetric scans ($500^2 \times 50$) of brain vessels in two channels, similar to settings in [3] we use 8 3D images for training and 3 for testing, using both input channels simultaneously. Our CLD contains volumetric scans ($1024^2 \times 600$) of bone cement lines, we use 13 3D images for training and 4 for testing. We conduct a five-fold cross-validation for all datasets.

We further note that the high-resolution of HRF-Retina is especially suited for examining the influence of the critical pixel mask as the topological errors are represented by more pixels than in lower-resolution images.

A.6. Compared Methods

The nnU-Net [10] is a framework that provides an improved version of the standard U-Net [23] with optimized hyperparameters. The nnU-Net is by default trained with a compound loss, consisting of Dice and Cross-Entropy loss [1].

clDice [3] is another topology-aware segmentation loss based on critical pixels that focuses on the extracted skeleton of likelihood maps and ground truth masks.

Method optimization. To demonstrate the effectiveness and better topological performances of our proposed method, we ensure identical prerequisites of our proposed method and the compared methods so that better performance can be clearly attributed to our proposed critical pixel mask. In that spirit, we conduct hyperparameter optimizations not only for our proposed method but also for the compared methods. For compound clDice, we conducted a weight hyperparameter search for each dataset ranging from 0.1 to 0.5 in 0.1 steps. For space reasons, we only report the compound clDice result with the highest clDice metric in Tab. 1. For clDice, we use the weight with the highest clDice metric for each dataset from their paper. For CLD, we choose the same weight for clDice as for the optimized compound clDice.

Omitted Methods. We compare against clDice loss [3], a state-of-the-art topology-aware segmentation loss, which defines the skeleton of predictions and ground truth as critical pixel mask. clDice and [8] only differ in L_{pixel}. clDice and [4] only differ in the skeleton algorithm, having the same L_{pixel} and critical pixel mask. [5] is the multi-class adaption of clDice with a different L_{pixel} and a reduced critical pixel mask in comparison to clDice that only includes the slightly dilated ground truth skeleton for computational efficiency. Additionally, [5] removes small structures during training which makes it difficult to point out from which design-choice their reported improvement over clDice (no removed structures) originates. However, as [5] note themselves an inferior skeleton extraction for binary segmentation compared to clDice, we suspect their critical pixel mask selection to be inferior to clDice in binary segmentation as it doesn't consider false positive connections.

Our proposed method can be adapted with any skeleton extraction and L_{pixel}. Hence, we focus on validation against methods with a distinctly different critical pixel mask selection strategy. Therefore, we don't validate against [4, 8] and [5], which use different skeletonization methods or L_{pixel} than clDice but not superior critical pixel masks. CLoss outperforming compound clDice implies CLoss outperforming [4, 5, 8].

We would have liked to compare against [7] but weren't able to reproduce their results. We could only reproduce their proposed critical pixel mask for the 2D datasets but not on our 3D CLD. We suspect that this might be caused by the complex 3D surface structure in our CLD. Initial training attempts with the original code from their repository resulted in numerous errors for the 2D and 3D datasets, which we could only fix for the 3D datasets. However, related work [5, 8, 13, 14] also doesn't report results on [7], so the implementation of [7] might be a common issue.

There also exist numerous other topology-preserving approaches that we don't compare against, since this would go well beyond the scope of this paper. These other approaches are not based on critical pixel masks and include for example graph-based approaches [13] ([13] only works with 2D data) or approaches entirely based on post-processing [14] (omits to improve topology correctness already in the image domain). We note that [14] can be complemented with our proposed method.

A.7. Evaluation Metrics

Evaluation patch sizes. We evaluated the Dice metric respectively over the whole test samples for the used 2D and 3D datasets.

As mentioned in Sec. 2.2, e_0 is not able to capture gaps if it is computed over the whole image for datasets with high overall connectivity. We calculate the metrics on patch sizes which provide meaningful evaluation for the used datasets. We iterate over the whole image shape with the patches as sliding window (similar to [8]) instead of sampling patches randomly for evaluation like [3, 6, 7, 13].

For 2D datasets, we evaluate clDice and AGS on the whole image. We calculate betti-metrics and e_0-Gt on patches, 375^2 (Roads) and 292^2 (HRF-Retina).

For the 3D dataset Vessap, we calculate clDice, AGS and e_0-Gt over the whole sample volume. We calculate betti-metrics on full image size along the z-direction.

For our 3D CLD, we calculate clDice, AGS and e_0-Gt over patches of $1024^2 \times 64$, where we also consider corrections of the clDice metric for its shortcoming with empty patches. We note that this patching in comparison to Vessap is due to memory reasons. Further, we evaluate betti-metrics on full image size along the z-direction.

A.8. Implementation Details

We conduct all our trainings within the nnU-Net framework [10] to ensure a maximum of reproducibility and leading performance [22]. The nnU-Net standard training has a length of 1000 epochs and utilizes Dice and Cross-Entropy as equally weighted compound loss. Training is done with a five-fold cross-validation for all methods. Inference is by standard done with all five folds simultaneously as an ensemble.

For our 2D datasets, we use the nnU-Net configuration 2d. For our 3D datasets, we use the configuration 3d_fullres.

We used the standard nnU-Net configuration for all methods, meaning our results can be reproduced with any GPU that has more than 11 GB VRAM. For increased speed, we conduct all trainings on A100 GPUs. As our used datasets are comparably small, we use the standard nnU-Net configuration without the residual encoder presets. We use PyTorch framework version 2.3.0 to implement our proposed method.

The nnU-Net configured a batch size of two for all datasets. The patch sizes for the 2D datasets were configured by nnU-Net to 1280 × 1024 (Roads), 1536 × 1024 (HRF-Retina) and for the 3D datasets to $256^2 \times 32$ (Vessap) and $160^2 \times 90$ (CLD).

For connectivity, we apply 8-adjacency in 2D and 26-adjacency in 3D.

Inference. Alg. 2 displays the pseudo-code for the inference with our proposed training-regime from Fig. 2.

Algorithm 2 CLoss inference

Input: D % domain image

$X_{bin}^{pre} \leftarrow predictions_{Pretraining}(D)$ % Binarized

$X_{bin}^{ft} \leftarrow predictions_{fine-tuning}(D)$ % Binarized

$X_{bin} \leftarrow post\text{-}processing(X_{bin}^{pre}, X_{bin}^{ft})$
% Only keeps structures of X_{bin}^{ft} already present in X_{bin}^{pre}

Output: X_{bin} % Binarized final predictions

Guideline for tuning gamma. If γ tends to 0, CLoss will converge to Dice&CE loss. A higher γ encourages more topological correctness. It should be $\gamma < 0.5$ to not focus overly on topological errors. All our datasets show good results for a value of 0.2. For new datasets, we recommend the common exploration approach in multiples of 3, e.g. 0.3, 0.1, 0.03.

Comment on generalization of CLoss to object segmentation of tumours and organs. CLoss promotes topology-preserving segmentation along the skeleton-pixels of the predictions and ground truth. Therefore, CLoss can be beneficial in all areas where the skeleton-pixels yield desirable information. We note that skeleton-pixels are typically not important for spherical objects like the human heart but are of increased importance for elongated objects like the trachea.

A.9. Additional Discussion of Quantitative Results

HRF-Retina. CLoss has the smallest error on the number of connected components (e_0) in all datasets except for HRF-Retina. We attribute the value for HRF-Retina to the mentioned susceptibility of e_0 to artifacts (Sec. 2.2) from the patch-based evaluation. The artifact distortion of e_0 is indicated by e_0-Gt, AGS, and the clDice metric. Our proposed metrics e_0-Gt and AGS clearly show a better gap closing of CLoss. Additionally, also the clDice metric of CLoss for HRF-Retina is significantly improved (about 2%) over its closest competing method. The clDice metric is especially reliable for HRF-Retina due to the smooth surface structure of the vessels. This indicates a better topology performance of CLoss for HRF-Retina despite the seemingly unfavorable e_0 value.

Vessap. For the Vessap dataset, we observe that the best clDice score is achieved by the topology-insensitive nnU-Net (pretraining). This seems to contradict with the other topology-sensitive metrics. Therefore, we suspect that the thin network 3D structures of Vessap could lead to many seemingly false positive predictions of the prediction skeleton, which distorts the topological sensitivity of the clDice metric on this dataset, see Fig. 4. This hypothesis is supported by an increasing AGS score for the topological fine-tuned methods. We also note that the seemingly high e_0 values for Vessap come from our patch-based evaluation of (full 2D image sizes stepping along the z-direction) in combination with the 3D network structure of Vessap. We note that e_1 in this context can especially contain artifacts from the slicing. Therefore, a minimized e_0 indicates topological performance on Vessap more reliable than minimized e_1.

CLD. The unmodified clDice loss only has a better gap closing (better e_0-Gt and AGS) than our standard CLoss on CLD. Importantly, this is only due to a different pixel-wise loss and not due to the different critical pixel mask. We verify this by adding results of CLoss (Dice) to CLD, which has the same pixel-wise loss function as clDice and only differs in the critical pixel mask to the clDice loss. CLoss (Dice) visibly outperforms clDice on the topology metrics. Interestingly, CLoss (Dice) is better than our proposed standard CLoss in e_1, e_0-Gt, and AGS but not in clDice, e, and e_0. This indicates that the target structure of our CLD favors different L_{pixel} for different metrics, which is not the case for the other datasets (compound clDice consistently better or comparable to original clDice). The favoring of different L_{pixel} for different metrics further illustrates the increased complexity of CLD over the other datasets.

A.10. Ablations

Post-Processing. Additional results of the post-processing are displayed in Tab. 2. All metric values in this example improve except for e_1, which remains the same, and AGS, which slightly decreases. All methods benefit from our post-processing in a similar quantity. It can be observed for all results, that the change in the Dice value is comparably small to the change in e. This indicates, that our post-processing primarily removes noise in the form of small separated structures.

Fine-tuning length. Ablations for the fine-tuning length are displayed in Tab. 3. A shorter fine-tuning length seems preferable, although we note that there is no consistent trend between the different lengths. For more detailed ablations on the fine-tuning between 50 epochs and 100 epochs, there was no clear winner, as might be falsely suggested by Tab. 3. Hence, we chose 50 epochs for computational efficiency.

Critical pixel mask. Additional ablations for the critical pixel mask are included in Tab. 4. We include Thin CLoss, which has a critical pixel extraction analog to CLoss except for the context extraction. Therefore, Thin CLoss only considers the skeleton at the topological errors. Our proposed critical pixel mask with context extraction yields significantly better topological correctness. CLoss seems only second best in e_1 to compound clDice ($\gamma = 0.1$), but the other bad metric values of compound clDice ($\gamma = 0.1$) indicate that this is likely not due to overall topological correctness but rather related to artifacts.

Pixel-wise loss. An ablation on L_{pixel} is included in Tab. 5. clDice and compound clDice differ only in L_{pixel}. clDice has L_{Dice}, and compound clDice the equally weighted combination of L_{Dice} and L_{CE}, analogue to our CLoss implementation. Compound clDice performs better on average for Roads, HRF-Retina, and Vessap. For CLD we note an advantage for clDice.

Compound clDice optimization. Additional results of the compound clDice optimization are displayed in Tab. 6. CLoss yields consistently better results.

Runtime. Results on the runtime are displayed in Tab. 7. The runtime of CLoss is comparable to clDice. In perspective, other state-of-the-art approaches [6,7,13] have 10 to 60 times longer loss calculations than clDice, as compared by [13]. Generating predictions on the test set is independent of the utilized loss and takes about 3min for the Roads dataset.

Table 2: **Post-Processing.** Example for CLD.

Post-Processing	Method	Weight γ	Dice↑	clDice↑ [3]	e↓	e_1↓	e_0↓	e_0-Gt↓	AGS↑
w/o	Dice & CE		70.23	85.17	3.615	1.075	2.540	2.479	82.03
with			70.27	85.23	3.422	1.075	2.347	2.447	82.02
w/o	Compound clDice	0.5	70.58	85.39	3.325	1.065	2.259	2.176	82.67
with			**70.61**	85.45	3.165	1.065	2.099	2.150	82.66
w/o	*CLoss*	*0.2*	70.41	86.75	3.224	**1.047**	2.177	1.916	**85.99**
with			70.44	**86.83**	**3.020**	**1.047**	**1.973**	**1.899**	85.98

Table 3: **Fine-tuning length.** Example for CLD. All results are post-processed.

Epochs	Method	Weight γ	Dice↑	clDice↑ [3]	e↓	e_1↓	e_0↓	e_0-Gt↓	AGS↑
50	Compound clDice	0.5	70.61	85.45	3.165	1.065	2.099	2.150	82.66
100			70.67	85.37	3.108	1.062	2.046	2.086	82.79
150			70.58	85.22	3.097	1.064	2.034	2.180	82.19
300			70.65	85.44	3.115	1.053	2.062	2.097	82.71
50	*CLoss*	*0.1*	70.75	**86.26**	3.109	1.069	2.040	2.040	84.44
100			**70.77**	86.25	**3.046**	**1.050**	**1.996**	**2.010**	**84.78**
150			70.46	85.64	3.173	1.062	2.110	2.218	83.02
300			70.51	85.34	3.075	1.060	2.015	2.079	82.39

Table 4: **Critical pixel mask.** Example for CLD. All results are post-processed. All methods differ only in the critical pixel mask.

Critical pixel mask	Method	Weight γ	Dice↑	clDice↑ [3]	e↓	e_1↓	e_0↓	e_0-Gt↓	AGS↑
w/o	Dice & CE		70.27	85.23	3.422	1.075	2.347	2.447	82.02
Full skeleton	Compound clDice	0.1	68.93	83.11	3.522	**1.029**	2.493	2.679	79.06
		0.2	70.47	84.96	3.361	1.077	2.283	2.352	81.81
		0.3	70.61	85.41	3.230	1.077	2.153	2.249	82.80
		0.4	70.51	85.16	3.287	1.063	2.223	2.283	82.26
		0.5	70.61	85.45	3.165	1.065	2.099	2.150	82.66
Skeleton at topological errors	*Thin CLoss*	0.08	70.33	85.06	3.353	1.087	2.266	2.438	81.53
		0.2	69.76	84.39	3.356	1.095	2.261	2.492	79.66
		0.5	69.62	84.05	3.475	1.091	2.383	2.643	79.29
Context at topological errors	*CLoss*	*0.08*	**70.98**	86.22	3.158	1.063	2.095	2.061	84.47
		0.1	70.75	86.26	3.109	1.069	2.040	2.040	84.44
		0.2	70.44	**86.83**	**3.020**	1.047	**1.973**	**1.899**	**85.98**

Table 5: **Pixel-wise loss.** We focus on ablation between L_{Dice} and our L_{pixel} used for CLoss. All results are post-processed.

Dataset	Method	Weight γ	Dice↑	clDice↑ [3]	e↓	e_1↓	e_0↓	e_0-Gt↓	AGS↑
Roads	clDice [3]	0.5	79.50	89.11	1.126	0.897	0.230	0.710	87.07
	Compound clDice	0.5	**79.59**	**89.21**	**1.065**	**0.870**	**0.195**	**0.673**	**87.25**
HRF-Retina	clDice [3]	0.5	82.15	**83.09**	0.426	0.256	0.170	2.475	**82.23**
	Compound clDice	0.5	**82.33**	82.98	**0.405**	**0.250**	**0.155**	**2.429**	81.25
Vessap	clDice [3]	0.4	92.70	94.86	29.00	1.240	27.76	**9.44**	**97.74**
	Compound clDice	0.4	**92.98**	**95.42**	**26.880**	**1.220**	**25.660**	11.040	97.28
CLD	clDice [3]	0.5	**70.88**	**86.22**	3.374	**1.042**	2.333	**1.825**	**87.11**
	Compound clDice	0.5	70.61	85.45	**3.165**	1.065	**2.099**	2.150	82.66

Table 6: **Compound clDice optimization.** Example for Roads dataset. All results are post-processed.

Method	Weight γ	Dice↑	clDice↑ [3]	e↓	e_1↓	e_0↓	e_0-Gt↓	*AGS*↑
nnU-Net [10]		79.69	89.34	1.181	0.895	0.286	0.702	86.46
Dice & CE		79.77	89.37	1.156	0.949	0.207	0.699	86.41
Compound clDice	0.1	79.69	89.26	1.089	0.866	0.223	0.691	86.78
	0.2	79.59	89.12	1.082	0.880	0.202	0.708	86.87
	0.3	79.73	89.34	1.108	0.909	0.199	0.673	87.05
	0.4	79.68	89.26	1.125	0.915	0.210	0.677	87.05
	0.5	79.59	89.21	1.065	0.870	0.195	0.673	87.25
CLoss	*0.08*	**79.82**	**89.47**	1.065	0.880	0.185	0.656	87.70
	0.1	79.57	89.21	**0.990**	0.810	**0.180**	0.617	87.88
	0.2	79.12	89.13	0.994	**0.788**	0.205	**0.494**	**89.48**

Table 7: **Runtime.** Example for Roads dataset.

Loss	Time/epoch	Time/training
Dice & CE	**22s**	**29min**
clDice	63s	59min
CLoss	78s	71min