Towards High-Quality Image Segmentation: Improving Topology Accuracy by Penalizing Neighbor Pixels

Supplementary Material

A. Implementation (code)

The three-lines implementation of SCNP that we employed in this study. Note that Y is the one-hot encoded ground truth, stride is always 1, and the padding is determined by the kernel size. Also note that SCNP only needs to be applied during training.

```
logits = model(X)
# SCNP Begin
t1 = -torch.nn.functional.max_pool2d(-(logits*Y +
     9999*(1-Y)), kernel_size=3, stride=1,
    padding=1)
t2 = torch.nn.functional.max_pool2d((logits*(1-Y)
     - 9999*Y), kernel_size=3, stride=1, padding
    =1)
scnp_logits = t1*Y + t2*(1-Y)
# SCNP End
loss = CEDiceLoss(softmax(scnp_logits), Y)
```

Listing 1. SCNP implementation.

B. Computational resources required by loss functions and SCNP

Tables B.1 and B.2 show the running time of one epoch and the GPU memory required by each loss function and our SCNP. These benchmarks were run on a cluster node entirely reserved for computing the measurements, consisting of 2x Intel(R) Xeon(R) Gold 6326 @ 2.90GHz and 1x NVIDIA A10 with 24GB. For SCNP, we computed the difference in resources when optimizing $\mathcal{L}_{CEDice}$ and $\mathcal{L}_{\overline{CEDice}}$.

We benchmarked all methods on FIVES (2D) and PulmonaryVA (3D). The size of the tensors sent to the model were $2 \times 2 \times 1280 \times 1024$ and $2 \times 3 \times 160 \times 128 \times 112$ on FIVES and PulmonaryVA, respectively. An epoch consisted on 300 training iterations; on FIVES that amounted to 2100 forward passes, whereas on PulmonaryVA it amounted to 1200 forward passes. This is due to the use of Deep supervision and different models that were automatically configured by nnUNetv2.

Loss	Epoch time (sec.)	GPU (MiB)
CEDice	54.85	5350
clDice ($i = 24$)	181.78	19340
SkelRecall	115.59	5334
TopoLoss	1505	5334
RWLoss	56.59	5778
Tversky ($\beta = 0.7$)	54.76	5326
Focal	55.26	5678
Extra resources required by:		
SCNP	+1.47	+340

Table B.1. Resources required on FIVES (2D nnUNetv2).

Loss	Iteration time (sec.)	GPU (MiB)
CEDice	72.79	6644
clDice ($i = 16$)	199.10	38672
SkelRecall	157.17	7784
TopoLoss	12960	6632
RWLoss	79.42	6864
Tversky ($\beta = 0.6$)	73.02	6644
Focal	70.65	6778
Extra resources required by:		
SCNP	+0.66	+454

Table B.2. Resources required on PulmonaryVA (3D nnUNetv2).

C. Gradients

The derivative of a loss function w.r.t. the logits can be obtained via the chain rule

$$\frac{\partial \mathcal{L}}{\partial z_{ki}} = \frac{\partial \mathcal{L}}{\partial \hat{y}_{ki}} \cdot \frac{\partial \hat{y}_{ki}}{\partial z_{ki}}, \tag{1}$$

where the first term is the derivative of the loss w.r.t. the sigmoid/softmax normalized predictions, and the second term is the derivative of the normalized predictions w.r.t. the unnormalized predictions (the logits $\mathbf{Z}$). Softmax normalization couples the gradients at a particular pixel location across all classes

$$\frac{\partial \hat{y}_{ki}}{\partial z_{ki}} = \hat{y}_{ki}(\delta_{ck,i} - \hat{y}_{ci}), \tag{2}$$

where $\delta_{ck,i}$ is the Kronecker delta, which is 1 if $k = c$ at pixel i and 0 otherwise. This is because each individual value will depend on the logits at that pixel location across all classes. Sigmoid normalization, in contrast, does not have this coupling effect, as sigmoid is applied individually and independently to each logit:

$$\frac{\partial \hat{y}_{ki}}{\partial z_{ki}} = \hat{y}_{ki}(1 - \hat{y}_{ki}). \tag{3}$$

The derivative of Cross entropy loss is

$$\frac{\partial \mathcal{L}_{CE}}{\partial \hat{y}_{ki}} = -\frac{y_{ki}}{\hat{y}_{ki}}, \tag{4}$$

where we can observe that the gradient at a pixel location depends on the values of the prediction and the ground truth at that pixel. The derivative of Focal loss is

$$\frac{\partial \mathcal{L}_{Focal}}{\partial \hat{y}_{ki}} = \alpha y_{ki} \left[\gamma (1-\hat{y}_{ki})^{\gamma-1} \log(\hat{y}_{ki}) - \frac{(1-\hat{y}_{ki})^{\gamma}}{\hat{y}_{ki}} \right], \tag{5}$$

where α weighs the overall contribution of Focal loss, and γ controls the penalization over uncertain predictions. As a loss function modified from Cross entropy, the gradients provided by Focal loss at a particular pixel also depend on the pixel and the ground truth itself, plus on the two other hyper-parameters (α, γ). The derivative of Dice loss is

$$\frac{\partial \mathcal{L}_{Dice}}{\partial \hat{y}_{ki}} = -\frac{2}{C B_k^2}(y_{ki} B_k - A_k), \tag{6}$$

where C is the number of classes, and B_k and A_k are the union and intersection, respectively, between predictions and ground truth of class k. In Equation (6), we can observe that the gradients from Dice loss (similarly to other metric-based loss functions such as Tversky loss [42]) do not depend on the pixel prediction but on the ground truth at the pixel itself and on global statistics involving the number of true/false positives and true/false negatives.

Note how optimizing jointly Cross entropy and Dice loss applied to softmax probability values, which is the most common approach in domains such as in medical image segmentation, couples 1) pixel predictions across all classes (*i.e.*, $\hat{y}_{ci} \forall c \in \{1, \dots, C\}$), 2) their ground truth (y_{ki}), and 3) global statistics (A, B in Eq. (6)). Optimizing a loss function with the SCNP logits will, in addition, couple predictions from neighboring pixels across all classes by adding the following term to the chain rule:

$$\frac{\partial \tilde{z}_{ki}}{\partial z_{ki}} = \sum_{j \in \Omega(i)} \mathbf{1}[z_{ki} = \tilde{z}_{kj}] \frac{\partial \mathcal{L}}{\partial \tilde{z}_{kj}}, \tag{7}$$

where $\mathbf{1}[\cdot]$ is 1 if the inner condition is true and 0 otherwise.

C.1. Example

Let us compute the gradients at the pixel corresponding to the yellow logits (Figure 3, Paper). With $\frac{\partial \mathcal{L}_{\overline{CE}}}{\partial \tilde{z}_{ki}} = \hat{y}_{ki} - y_{ki}$, the gradients at ki are:

$$\begin{aligned} \nabla_{z_{ki}} \mathcal{L}_{CE} = & [(.50-1)+(.56-1), \\ & (.23-0)+(.21-0)+(.19-0), \\ & (.20-0)+(.24-0)] \\ = & [-.94, .63, .44] \end{aligned} \tag{8}$$

D. SCNP's negligible delay in convergence

Since SCNP switches the focus of the optimization from improving all pixels to improving the worst neighbors, we investigated whether and to what extent SCNP delays convergence. Figure D.1 shows, in two loss functions (Cross Entropy and Dice loss), 1) that the delay in convergence due to optimizing the SCNP logits alone is negligible, and 2) that such delay can be further reduced by optimizing the SCNP logits jointly with the standard logits.

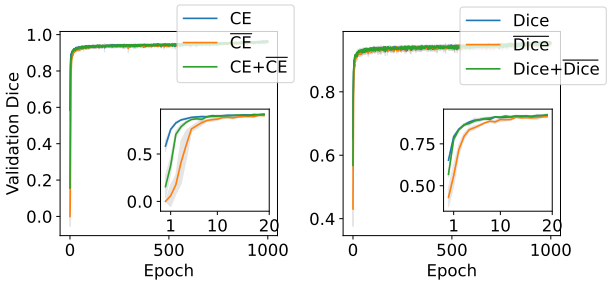


Figure D.1. Left: Convergence plot when optimizing Cross entropy over the standard logits, the SCNP logits, and both. Right: Same scenario with Dice loss. The small plots inside the two panels are a zoomed in on the first 20 epochs.

E. SCNP vs. Binary closing

We investigated the effectiveness of binary closing (•) by applying it to CEDice segmentations on four datasets (one per group). We used kernels of size 3, 5, 7 (2D & 3D), and we applied binary closing until convergence. Below, we report the best β_{0e} (*i.e.* accuracy in the number of structures), which indicates topology accuracy.

Dataset	CEDice	•(CEDice)	**SCNP** ($\overline{CEDice}$)
FIVES	10.82$\pm$2.71	8.79$\pm$2.47	**6.94**$\pm$**1.47**
DeepGlobeRoads	11.77$\pm$3.17	9.92$\pm$2.83	**6.56**$\pm$**2.25**
CirrMRI600	0.27$\pm$0.26	0.19$\pm$0.13	**0.12**$\pm$**0.05**
NuInsSeg	8.18$\pm$2.39	8.17$\pm$2.40	**7.53**$\pm$**2.20**

Table E.3. Average β_{0e} in four datasets when optimizing CEDice (second column) and after applying binary closing (third column), and when optimizing CEDice with the SCNP logits (fourth column). •: Binary closing.

Although binary closing improved β_{0e}, SCNP remained superior. Furthermore, note that binary closing is a hurdle or ineffective in instance segmentation, real-time applications, tubular structures that are very close, and it requires a ground truth to find the optimal kernel size and number of iterations.

F. Comparison with clCE

We run extra experiments with [2]'s loss on FIVES, as FIVES was also used in [2]. clCE loss achieved: Dice 0.93$\pm$0.00, β_{0e} 10.62$\pm$3.05, clDice 0.92$\pm$0.00, with our SCNP remaining more effective (same Dice, clDice, β_{0e}: **6.94**$\pm$**1.47**).

Figure F.2 shows a visual comparison in two representative images.

G. Datasets description

Tables G.4 to G.7 list the details of the datasets used in this study, including an image/slice of each dataset and their size and split.

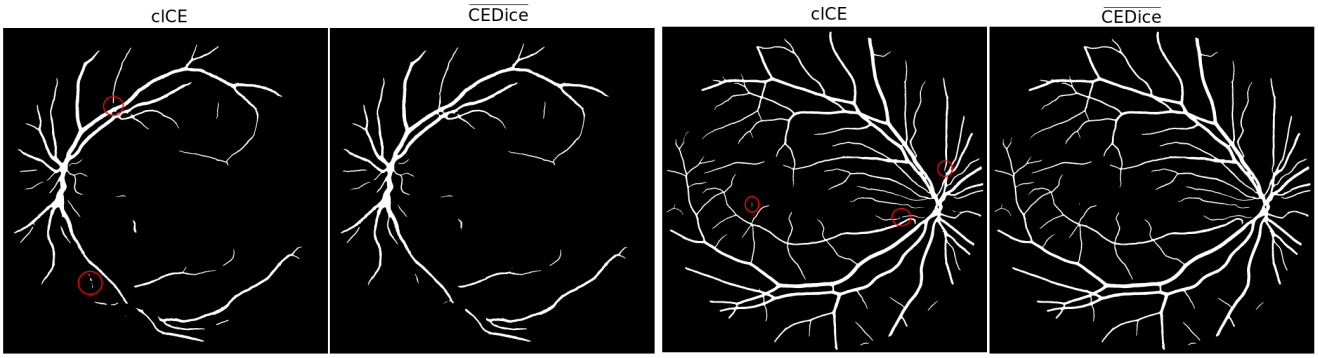


Figure F.2. Visual comparison between clCE and $\overline{CEDice}$. Red circle: topological errors.

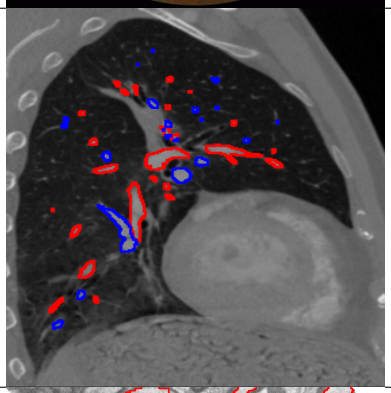
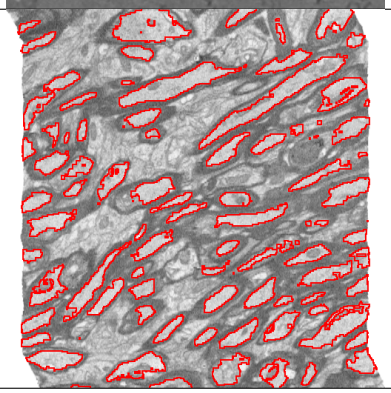
FIVES		• Blood vessels in fundus retina images • Semantic segmentation • R,G,B • 2048 × 2048 pixels	Splits: 480-120-200 (train, validation, test)
PulmonaryVA		• Pulmonary arteries and veins in chest CT scans • Semantic segmentation • CT • 184 × 235 × 262.5 voxels (median)	Dataset size: 106 (five-fold cross validation)
Axons		• Axons in mouse brain • Semantic segmentation • Electron-microscopy • 1062.5 × 1067 × 285 voxels (median)	Dataset size: 10 (five-fold cross validation) Sham_HM_25_contra—ipsi, Sham_HM_49_contra—ipsi, TBI_HM_24_contra—ipsi, TBI_HM_28_contra—ipsi, TBI_HM_2_contra—ipsi

Table G.4. Medical datasets with tubular structures (nnUNetv2 framework, Group ①)

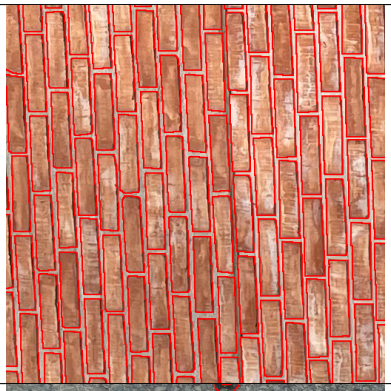
TopoMortar		• Mortar in red brick wall images • Semantic segmentation • R,G,B • 512 × 512 pixels	Splits: 50-20-350 (train, validation, test)
Crack500		• Cracks in concrete images • Semantic segmentation • R,G,B • 640 × 896 Pixels (median)	Splits: 250-50-200 (train, validation, test)
DeepGlobeRoads		• Roads in satellite images • Semantic segmentation • R,G,B • 1024 × 1024 pixels (median)	Splits: 4357-623-1246 (train, validation, test)

Table G.5. Non-medical datasets with tubular structures (Detectron2+DeepLabv3+ framework, Group ②)

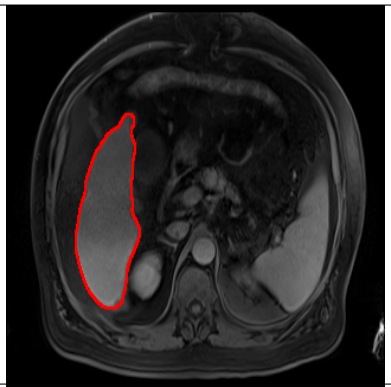
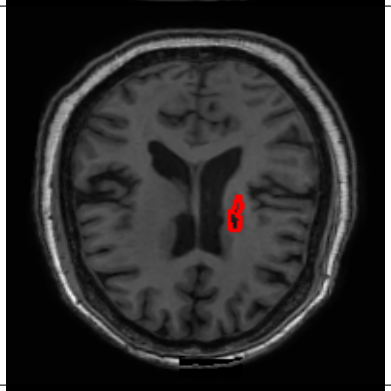
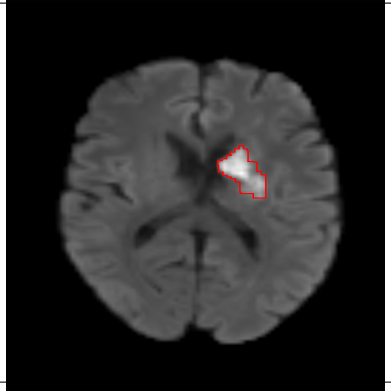
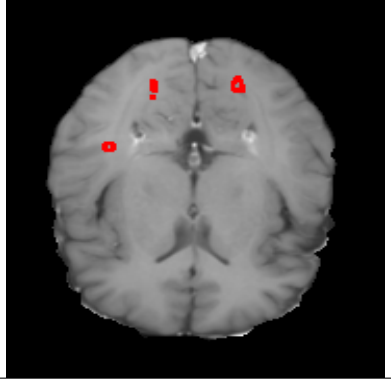
CirrMRI600		• Cirrhotic liver in MRI • Semantic segmentation • T2w • 320 × 260 × 80 voxels (median)	Splits: 198-50-31 (train, validation, test)
ATLAS2		• Lesion in brain MRI • Semantic segmentation • T1w • 197 × 233 × 189 voxels (median)	Dataset size: 655 (five-fold cross validation)
ISLES24		• Ischemic lesion in brain MRI • Semantic segmentation • ADC, DWI • 512 × 581 × 69 voxels (median)	Dataset size: 149 (five-fold cross validation)
MSLesSeg		• Multiple sclerosis lesion in brain MRI • Semantic segmentation • T1w, T2w, FLAIR • 182 × 218 × 182 voxels (median)	Splits: 74-19-22 (train, validation, test)

Table G.6. Medical datasets with non-tubular structures (nnUNetv2 framework, Group ③)

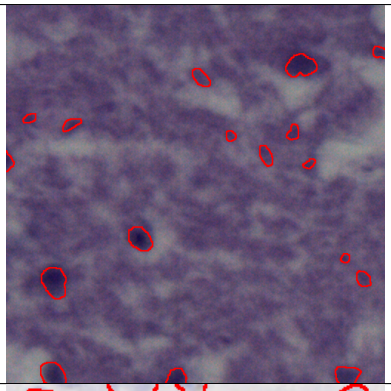
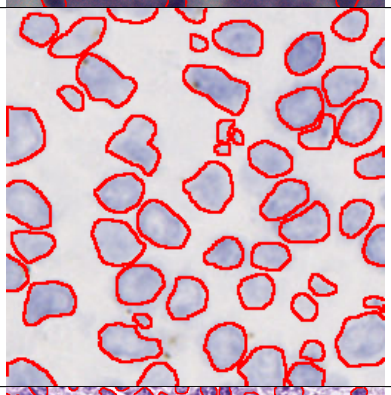
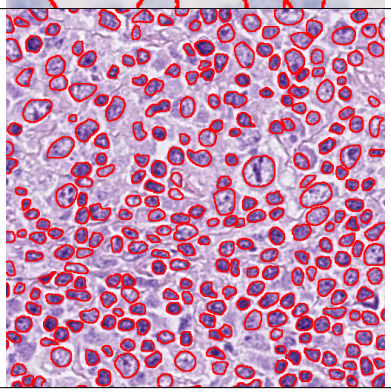
NuInsSeg		• Cell nuclei in H&E-stained histological images • Instance segmentation • R,G,B • 512 × 512 pixels (median)	Splits: 532-66-67 (train, validation, test)
IHC_TMA		• Cell nuclei in immunohistochemical tissue microarray images • Instance segmentation • R,G,B • 256 × 256 pixels (median)	Splits: 212-27-27 (train, validation, test)
LyNSeC		• Lymphoma nuclei in H&E-stained histological images • Instance segmentation • R,G,B • 512 × 512 pixels (median)	Splits: 559-70-70 (train, validation, test)

Table G.7. Medical (cells) datasets with rounded structures (InstanSeg framework, Group ④)

H. Optimization

We left the frameworks' optimization strategy as it was by default.

nnUNetv2 employs SGD with Nesterov momentum (0.99), a weight decay of 3e-5, initial learning rate of 0.01, and the following learning rate scheduler: $\eta_{it} = \eta_0(\frac{1-it}{iterations})^{0.9}$. Models were optimized for 1000 epochs.

Detectron2 employs AdamW [29] with an initial learning rate of 0.001, $\beta_1 = 0.9, \beta_2 = 0.999$, a weight decay of 0.1, and cosine annealing as learning rate scheduler. Models were optimized for 12000 iterations.

InstanSeg employs Adam [23] with an initial learning rate of 0.001. Models were optimized for 500, 150 and 500 epochs in NuInsSeg, IHC_TMA, and LyNSeC, respectively.

H.1. Loss functions and their hyper-parameters

Cross Entropy Dice loss (CEDice) We summed Cross Entropy and Dice loss: $\mathcal{L}_{CEDice} = \mathcal{L}_{CE} + \mathcal{L}_{Dice}$.

clDice loss We employed the official code provided by Shit et al. [44] and, as in the original study, we combined clDice with Dice loss. Since clDice's effectiveness depends on the suitability of the hyper-parameter "number of iterations" (i) of its soft-skeletonization method, we conducted hyper-parameter optimization. For each dataset, we explored five different values that were selected based on the thickness of the tubular structures while limiting the GPU memory usage to 80GB. The values that we explored in each dataset (and the **selected** hyper-parameter) were:

- FIVES: 8, 16, **24**, 32, 40.
- PulmonaryVA: 4, 8, 12, **16**, 18.
- Axons: 9, 18, 27, 36, **46**.
- TopoMortar: **4**, 8, 12, 16, 18.
- Crack500: **10**, 20, 30, 40, 51.
- DeepGlobeRoads: 15, 30, **45**, 60, 72.

Skeleton Recall (SkelRecall) We employed the official code provided by Kirchhoff et al. [24] and, as in the original study, we combined SkelRecall with Cross Entropy loss.

TopoLoss We employed the official code provided by Hu et al. [17]. As suggested in the original work, we first optimized Cross Entropy loss and, at the end of the optimization, we combined it with TopoLoss. Since TopoLoss computes persistence homology, the iteration time increased dramatically, especially for the 3D datasets. Therefore, we set the number of iterations that we optimized TopoLoss such that the total training time did not exceed 24 hours. Table H.8 summarizes this.

RegionWise (RWLoss) We employed the official code provided by Valverde and Tohka [47].

Dataset	Eps. w/ TopoLoss	% of train time
FIVES	38 (1000)	44.20%
PulmonaryVA	3 (1000)	34.30%
Axons	5 (1000)	51.88%
TopoMortar	3600 (12000)	87.86%
Crack500	3600 (12000)	62.65%
DeepGlobeRoads	3600 (12000)	41.50%
CirrMRI600	5 (1000)	39.11%
ATLAS2	12 (1000)	37.16%
ISLES24	12 (1000)	63.35%
MSLesSeg	5 (1000)	36.81%
NuInsSeg	150 (500)	36.80%
IHC_TMA	15 (150)	34.41%
LyNSeC	150 (500)	53.72%

Table H.8. Second column: Epochs optimizing TopoLoss and total epochs. Third column: Percentage of training time optimizing TopoLoss relative to the total training time (average across the five runs).

Focal loss We employed MONAI's implementation [3] of Focal loss [27], using its default hyper-parameters.

Tversky loss We employed MONAI's implementation [3] of Tversky loss [42], and we optimized its hyper-parameters α, β which control the penalty for false positives and negatives.

I. Performance metrics

Tables I.9 to I.12 show the results obtained by CEDice (our baseline), CEDice optimizing the SCNP-penalized logits, their combination, and several loss function, on the 13 datasets used in this study. Furthermore, we assess the statistical significance of SCNP-derived performance metrics compared to the baseline CEDice with a permutation t-test based on 10,000 random permutations ($p < 0.05$).

For the datasets with a pre-defined train-test split (Appendix G), we report the mean and standard deviation on the same test-set images across five runs; for the remaining datasets, we report the mean and standard deviation across the five-fold cross validation runs. In consequence, the standard deviation in the first group of datasets indicates algorithmic variance while in the second group it indicates intra-dataset variance. Appendix J presents a qualitative comparison of segmentations.

	Loss	Dice	β_{0e}	clDice
FIVES	CEDice	0.92±0.0	10.82±2.71	0.92±0.0
	clDice ($i = 24$)	0.91±0.0	36.55±10.70	0.91±0.01
	SkelRecall	0.91±0.0	12.45±4.25	0.91±0.0
	TopoLoss	0.92±0.0	8.09±2.45	0.92±0.0
	RWLoss	0.92±0.0	15.99±5.35	0.92±0.0
	Tversky ($\beta = 0.7$)	0.92±0.0	12.22±3.45	0.92±0.0
	Focal	0.92±0.0	16.08±5.97	0.92±0.0
	CEDice+$\overline{CEDice}$	**0.93±0.0**	7.95±1.82	0.92±0.0
	$\overline{CEDice}$	0.92±0.0	**6.94±1.47**	0.92±0.0
PulmonaryVA	CEDice	0.84±0.04	12.89±12.92	0.88±0.04
	clDice ($i = 16$)	0.84±0.04	15.98±8.73	0.89±0.04
	SkelRecall	0.83±0.05	39.45±20.42	0.84±0.06
	TopoLoss	0.84±0.05	14.44±10.67	0.88±0.04
	RWLoss	0.84±0.05	18.02±13.83	0.87±0.04
	Tversky ($\beta = 0.6$)	0.84±0.05	15.44±14.87	0.88±0.04
	Focal	0.84±0.05	21.29±13.4	0.87±0.04
	CEDice+$\overline{CEDice}$	0.84±0.05	**5.82±5.02**	0.87±0.06
	$\overline{CEDice}$	0.83±0.05	**6.95±4.18**	0.84±0.07
Axons	CEDice	0.90±0.04	531±534	0.85±0.05
	clDice ($i = 46$)	0.89±0.04	1925±1701	0.86±0.04
	SkelRecall	0.89±0.04	1578±900	0.85±0.05
	TopoLoss	0.90±0.04	539±490	0.85±0.05
	RWLoss	0.90±0.04	408±412	0.85±0.05
	Tversky ($\beta = 0.6$)	0.90±0.04	530±623	0.85±0.05
	Focal	0.90±0.04	412±400	0.85±0.05
	CEDice+$\overline{CEDice}$	0.90±0.04	254±203	0.87±0.05
	$\overline{CEDice}$	0.90±0.04	279±207	0.88±0.05

Table I.9. Group ① Medical datasets with tubular structures (nnUNetv2). $\overline{CEDice}$: CEDice applied to SCNP. Bold (Dice): Highest and significantly significant. Bold (β_{0e}, clDice): Best scores with Dice equal or better and statistically significant.

	Loss	Dice	β_{0e}	clDice
TopoMortar	CEDice	0.67±0.03	25.06±9.26	0.75±0.03
	clDice ($i = 4$)	0.68±0.02	21.40±5.86	0.75±0.02
	SkelRecall	0.68±0.02	23.64±7.30	0.76±0.02
	TopoLoss	0.67±0.02	27.22±7.64	0.74±0.02
	RWLoss	0.54±0.30	19.40±13.80	0.76±0.03
	Tversky ($\beta = 0.8$)	0.68±0.02	22.59±6.46	0.75±0.02
	Focal	0.67±0.02	29.75±10.17	0.74±0.02
	CEDice+$\overline{CEDice}$	0.68±0.02	20.86±6.78	0.76±0.02
	$\overline{CEDice}$	0.68±0.02	20.30±6.45	0.75±0.02
Crack500	CEDice	0.68±0.03	10.61±3.44	0.76±0.03
	clDice ($i = 10$)	0.65±0.09	9.98±2.96	0.80±0.04
	SkelRecall	0.65±0.02	11.78±3.22	0.77±0.02
	TopoLoss	0.54±0.04	13.34±3.24	0.58±0.04
	RWLoss	—	—	—
	Tversky ($\beta = 0.6$)	0.69±0.02	10.47±3.67	0.77±0.03
	Focal	—	—	—
	CEDice+$\overline{CEDice}$	0.68±0.03	9.32±2.71	0.76±0.03
	$\overline{CEDice}$	0.66±0.03	9.48±2.36	0.73±0.03
DeepGlobeRoads	CEDice	0.75±0.02	11.77±3.17	0.82±0.02
	clDice ($i = 45$)	0.72±0.03	6.51±2.41	0.82±0.03
	SkelRecall	0.71±0.02	15.26±4.12	0.79±0.02
	TopoLoss	0.22±0.05	27.79±6.04	0.33±0.06
	RWLoss	—	—	—
	Tversky ($\beta = 0.7$)	0.74±0.02	9.78±2.90	0.82±0.02
	Focal	—	—	—
	CEDice+$\overline{CEDice}$	0.72±0.03	7.24±2.63	0.80±0.03
	$\overline{CEDice}$	0.70±0.03	6.56±2.25	0.78±0.03

Table I.10. Group ② Non-medical datasets with tubular structures (Detectron2+DeepLabv3+).

	Loss	Dice	β_{0e}
CirrMRI600	CEDice	0.97±0.01	0.27±0.26
	TopoLoss	0.97±0.01	0.32±0.28
	RWLoss	0.96±0.01	0.43±0.31
	Tversky ($\beta = 0.7$)	0.95±0.0	1.39±1.49
	Focal	0.97±0.01	0.51±0.30
	CEDice+$\overline{CEDice}$	0.97±0.01	0.17±0.09
	$\overline{CEDice}$	0.97±0.01	**0.12±0.05**
ATLAS2	CEDice	0.63±0.28	1.60±2.5
	TopoLoss	0.61±0.30	1.45±2.42
	RWLoss	0.56±0.33	1.44±2.47
	Tversky ($\beta = 0.8$)	0.61±0.26	2.37±3.15
	Focal	0.59±0.32	1.39±2.51
	CEDice+$\overline{CEDice}$	0.62±0.27	1.40±2.49
	$\overline{CEDice}$	0.61±0.28	1.38±2.57
ISLES24	CEDice	0.77±0.20	3.64±4.39
	TopoLoss	0.77±0.20	3.77±5.06
	RWLoss	0.77±0.20	3.92±5.03
	Tversky ($\beta = 0.7$)	0.77±0.19	4.61±7.65
	Focal	0.76±0.21	4.32±6.30
	CEDice+$\overline{CEDice}$	0.77±0.19	3.72±5.40
	$\overline{CEDice}$	0.76±0.20	5.27±7.85
MSLesSeg	CEDice	0.71±0.01	9.47±1.81
	TopoLoss	0.71±0.01	9.40±1.84
	RWLoss	0.70±0.01	10.22±1.64
	Tversky ($\beta = 0.8$)	0.71±0.01	9.55±2.14
	Focal	0.69±0.02	10.06±2.27
	CEDice+$\overline{CEDice}$	0.70±0.01	12.72±1.54
	$\overline{CEDice}$	0.67±0.01	23.05±1.22

Table I.11. Group ③ Medical datasets with non-tubular structures (nnUNetv2).

	Loss	Dice	β_{0e}	$(\mathcal{R}_e)$
NuInsSeg	CEDice	0.76±0.04	8.18±2.39	0.91±0.01
	TopoLoss	0.28±0.01	13.35±9.23	0.76±0.04
	RWLoss	0.72±0.04	8.42±3.04	0.90±0.01
	Tversky ($\beta = 0.6$)	0.77±0.03	7.94±2.91	0.90±0.01
	Focal	0.71±0.03	9.25±2.05	0.89±0.01
	CEDice+$\overline{CEDice}$	0.76±0.04	7.95±2.21	0.91±0.01
	$\overline{CEDice}$	0.76±0.03	7.53±2.20	**0.92±0.01**
IHC_TMA	CEDice	0.79±0.04	3.31±1.58	0.88±0.01
	TopoLoss	0.36±0.0	7.49±3.36	0.78±0.03
	RWLoss	0.78±0.05	2.33±1.36	0.87±0.01
	Tversky ($\beta = 0.9$)	0.81±0.04	2.64±1.32	0.88±0.01
	Focal	0.77±0.04	2.33±1.22	0.87±0.01
	CEDice+$\overline{CEDice}$	0.82±0.04	2.39±1.24	0.89±0.01
	$\overline{CEDice}$	0.81±0.04	2.81±2.04	**0.90±0.01**
LyNSeC	CEDice	0.88±0.0	10.23±3.94	0.91±0.0
	TopoLoss	0.47±0.0	34.44±19.91	0.73±0.05
	RWLoss	0.84±0.01	10.56±5.04	0.91±0.0
	Tversky ($\beta = 0.6$)	0.88±0.0	9.47±3.53	0.91±0.0
	Focal	0.86±0.01	10.10±3.92	0.90±0.0
	CEDice+$\overline{CEDice}$	0.88±0.0	9.75±3.42	**0.92±0.0**
	$\overline{CEDice}$	0.88±0.0	9.91±3.33	**0.92±0.0**

Table I.12. Group ④ Medical datasets with rounded structures.

J. Segmentation examples

Figures J.3 to J.6 illustrates the segmentations provided by our SCNP and other loss functions. For each group we show the segmentations obtained by optimizing CEDice loss and the top three loss functions.

K. Sensitivity analysis: extra metrics

Table K.13 and Table K.14 show the Dice and clDice metrics of the sensitivity analysis.

$\downarrow w$	9.69	7.23	4.70
3	$0.92_{\pm 0.0}$	$0.92_{\pm 0.0}$	$0.92_{\pm 0.0}$
5	$0.92_{\pm 0.0}$	$0.92_{\pm 0.0}$	$0.92_{\pm 0.0}$
7	$0.92_{\pm 0.0}$	$0.91_{\pm 0.0}$	$0.91_{\pm 0.0}$
9	$0.9_{\pm 0.0}$	$0.89_{\pm 0.0}$	$0.9_{\pm 0.0}$
11	$0.92_{\pm 0.0}$	$0.91_{\pm 0.0}$	$0.92_{\pm 0.0}$

Table K.13. Dice coefficients of the sensitivity analysis on SCNP's hyper-parameter w.

$\downarrow w$	9.69	7.23	4.70
3	$0.92_{\pm 0.0}$	$0.92_{\pm 0.0}$	$0.92_{\pm 0.0}$
5	$0.91_{\pm 0.0}$	$0.91_{\pm 0.0}$	$0.92_{\pm 0.0}$
7	$0.9_{\pm 0.0}$	$0.89_{\pm 0.0}$	$0.9_{\pm 0.0}$
9	$0.88_{\pm 0.0}$	$0.87_{\pm 0.0}$	$0.88_{\pm 0.0}$
11	$0.91_{\pm 0.0}$	$0.91_{\pm 0.0}$	$0.92_{\pm 0.0}$

Table K.14. clDice metric of the sensitivity analysis on SCNP's hyper-parameter w.

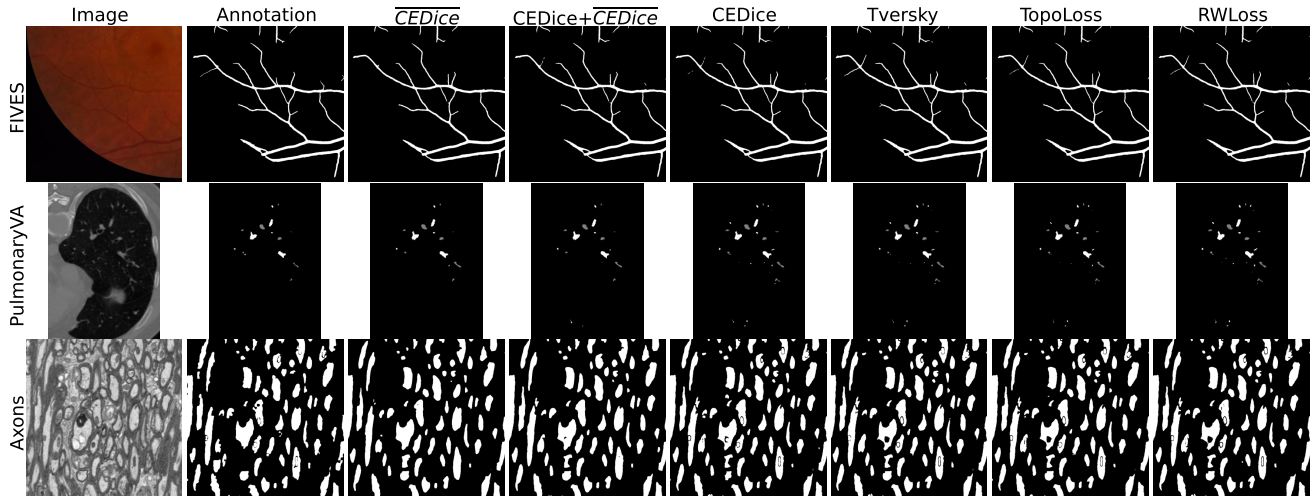


Figure J.3. Results on medical datasets with tubular structures (Group ①).

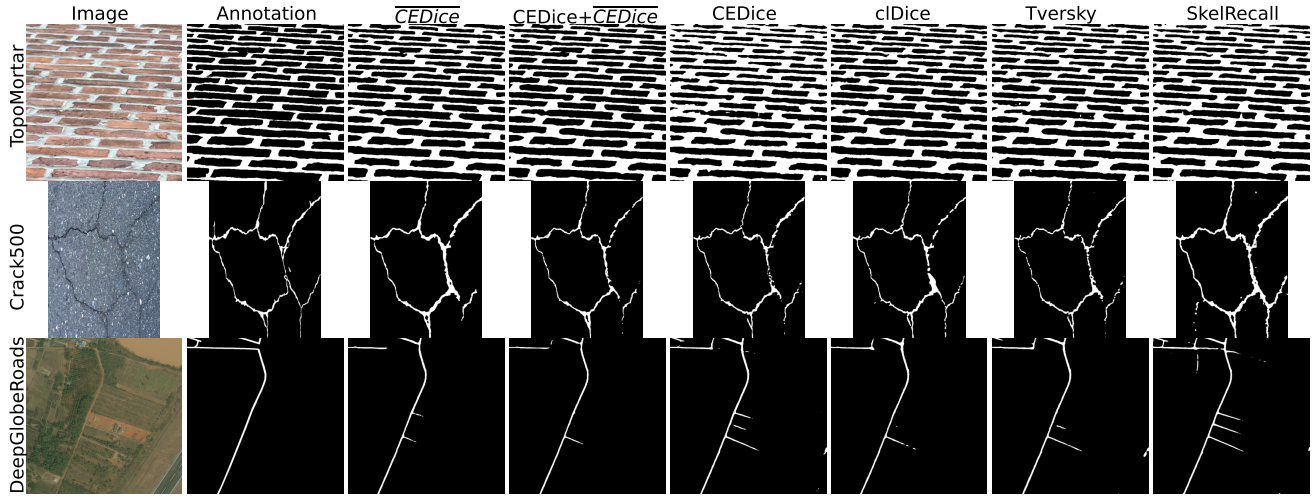


Figure J.4. Results on non-medical datasets with tubular structures (Group ②).

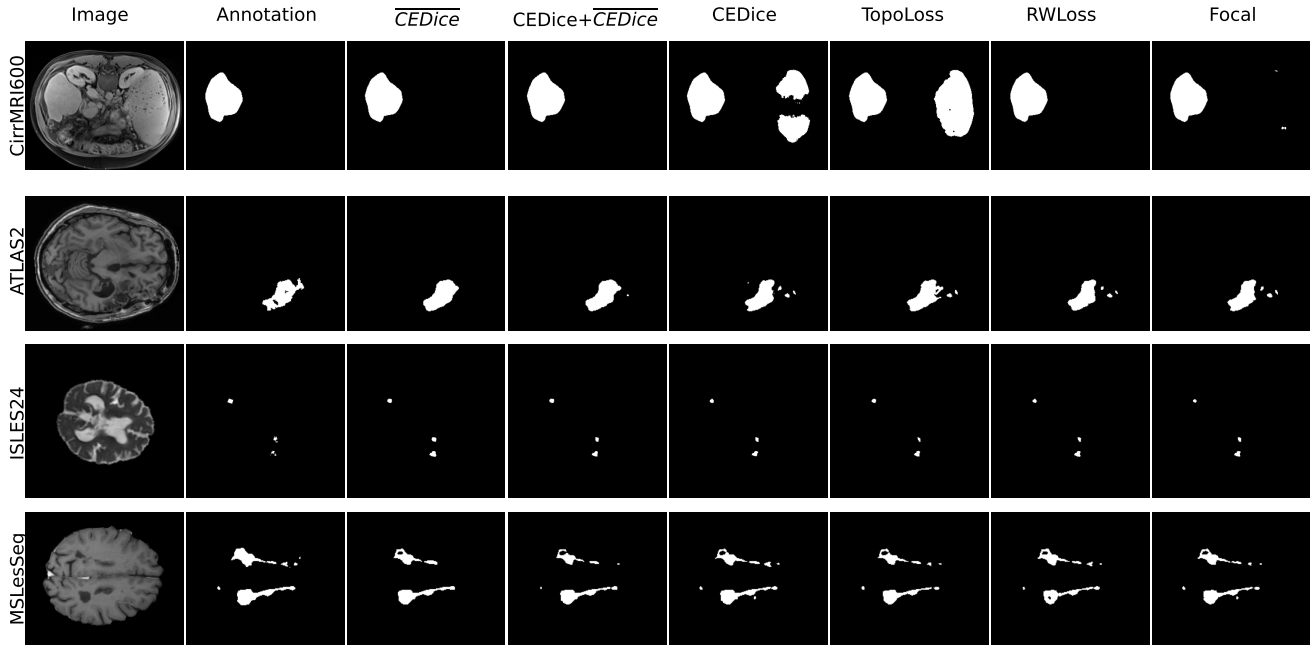


Figure J.5. Results on medical datasets with non-tubular structures (Group ③).

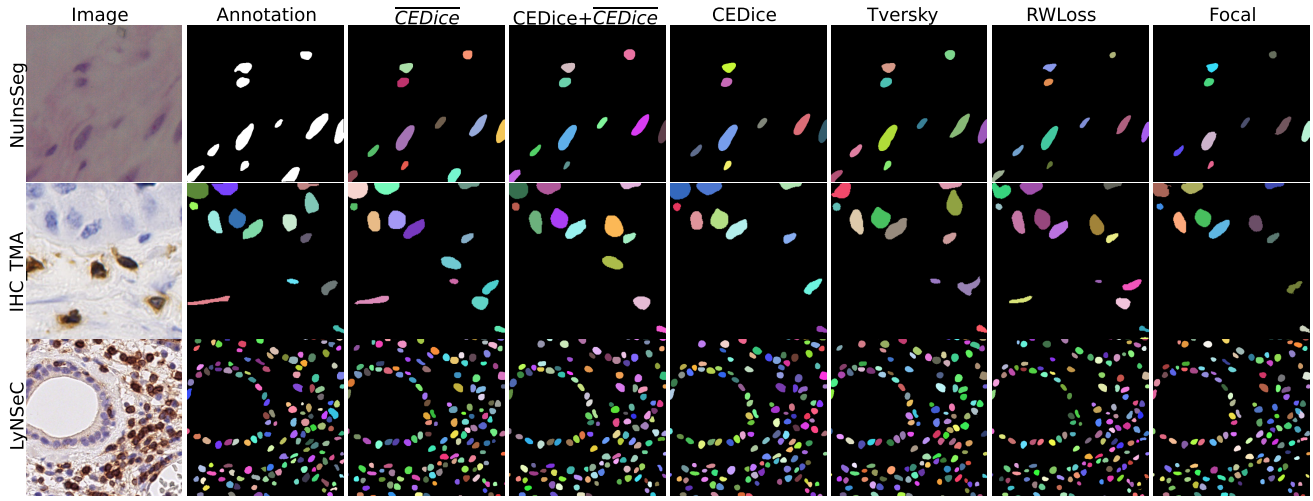


Figure J.6. Results on medical datasets with rounded structures (Group ④).