Learning Topological Interactions for Multi-Class Medical Image Segmentation — Supplementary Material —

Saumya Gupta[*], Xiaoling Hu[*], James Kaan, Michael Jin, Mutshipay Mpoy, Katherine Chung, Gagandeep Singh, Mary Saltz, Tahsin Kurc, Joel Saltz, Apostolos Tassiopoulos, Prateek Prasanna, and Chao Chen

Stony Brook University, Stony Brook NY 11794, USA
{saumya.gupta, xiaoling.hu, chao.chen.1}@stonybrook.edu

In the supplementary material, we begin with an alternate implementation of the naive method in Sec. 6. In Sec. 7, we provide illustrations of the connectivity kernel in both 2D and 3D settings. In Sec. 8-11, we provide detailed descriptions of the experiments, namely the datasets, architectures, implementations, additional ablation studies and qualitative and quantitative results.

6 Alternate Implementation of Naive Solution

In Sec. 3.1, we discussed the naive solution which involved simply looping over all the pixels and scanning all its neighbors. The obvious issue with loops is that though it takes $O(1)$ time to access the neighborhoods of any single pixel, it takes polynomial time to access the neighborhoods of all the pixels together. Here we discuss an alternate implementation of the same idea. Although it is more efficient than the naive method discussed in Sec. 3.1, it is still inferior to the convolution-based method in terms of speed and complexity. We provide details of the alternate naive method here in the supplementary due to space constraints in the main paper.

We continue to use the same terminology for terms A, B, C, P, d, k etc. as used in Sec. 3.1. We assume 2D 4-connectivity scenario.

Since the connectivity defined is constant for every pixel, we can translate the idea of looping over every pixel to that of *shifted* maps instead. A map P_r is obtained by shifting every pixel in P to the right by one pixel. Similarly we can obtain maps P_l, P_u and P_d, which are obtained from P by shifting one pixel to the left, up, and down, respectively. Thus for i, we have the 4-connectivity neighbors of the pixel $P[i]$, that is, $P_r[i]$, $P_l[i]$, $P_u[i]$, and $P_d[i]$. And with the help of these maps, we have access to the neighborhoods of every pixel simultaneously without loops. We can now use algebraic manipulation to determine whether pixel $P[i] \in A$ has a C neighbor(s) or not.

We need to prune the neighborhood maps so that we are left with only the critical pixel feature map. Let M_A be a mask obtained from P such that it contains a 1 at locations where the pixels are in A. Similarly, let M_C be a mask

[*] Equal contribution.

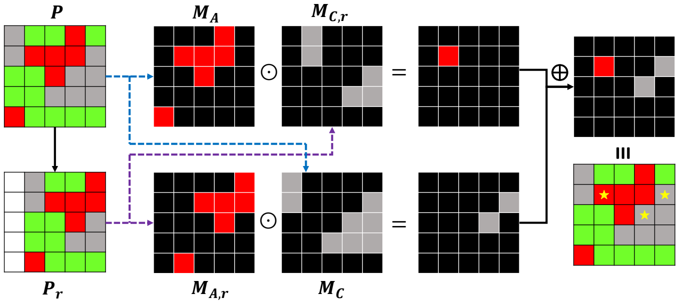


Fig. 8: 2D illustration of the **alternate naive** algorithm to detect the set of critical pixels. The figure demonstrates the logic to obtain the critical pixels in the right direction using P_r. The same logic needs to be extended to P_l, P_u, P_d to obtain the entire critical pixel set for the 4-connectivity case. Topologically critical pixels are marked with (*).

obtained from P for C. We similarly obtain masks $M_{A,w}$ and $M_{C,w}$ from each shifted map P_w. These masks reduce the context to classes A and C alone while discarding others. Note that here subscript w is used as a generic subscript to denote any of r, l, u, and d.

Now for each neighborhood map P_w, the term $(M_A \odot M_{C,w}) \cup (M_C \odot M_{A,w})$ gives the critical pixel map in that direction. Intuitively, it captures all the pixels of A that fall in the neighborhood of C and vice-versa. If we take the union of all these terms constructed from every direction, we obtain all the pixels in A and C which appear in each other's neighborhood. Fig. 8 gives an overview of the algorithm by obtaining the critical pixel map using only P_r. We can extend the same logic for other P_w. While intuitive, the disadvantage of this approach is in its scalablility with respect to d.

Computational Efficiency. We analyze the computational efficiency by determining complexity as a function of the input and neighborhood size. Let the image size be $N \times N$ and we enforce a separation of d pixels. In the alternate naive solution, we require d shifted maps along each direction, or $k = 2d$ maps along an axis. The time complexity is therefore in the order of $O(N^2k^2)$. The memory requirement will be $O(N^2)$ to store masks $M_{C,w}$ and $M_{A,w}$, and we can optimize this by using an allocated buffer into which we can keep over-writing the masks generated for each direction. As discussed in Sec. 3.1, the proposed solution has a time complexity of $O(N^2 \log N)$. Thus, the alternate naive solution is not scalable with respect to d (or k), whereas our proposed method has a running time independent of the specified neighborhood size. The memory requirements of both methods are similar. In practice, deep learning frameworks are highly optimized for convolution operations, and so they are much cheaper than computing shifts along axes.

Running Times. For the same network architecture, the inference time remains the same irrespective of the loss functions; the difference is in the training times. We further compare the training times of using the naive method, the alternate naive method and the convolution-based method in the topological interaction module. We report the average time for training one epoch on the IVUS dataset, having a batch size of 5, input size of 384 × 384, and $d = 1$. For the naive solution, it takes 69.4s to compute the L_{ti} for each epoch. With the alternate naive solution, it takes 5.9s to compute the L_{ti} for each epoch, while it takes only 0.8s for the proposed convolution-based method. The significant difference between the naive and convolution-based methods boils down to the fact that convolutions are highly optimized for GPUs, whereas looping across each pixel in CPU-space incurs huge time. We thus conclude that the convolution-based method is highly efficient compared to both the naive and alternate naive methods, and has negligible timing overhead.

7 Remark on the Connectivity Kernel K

In Fig. 9, we provide illustrations on how (for the same input) the critical pixels map (V) changes based on the connectivity kernel (K) used. We provide illustrations for the 2D case using 4-connectivity and 8-connectivity kernels, and, for the 3D case using 6-connectivity and 26-connectivity kernels.

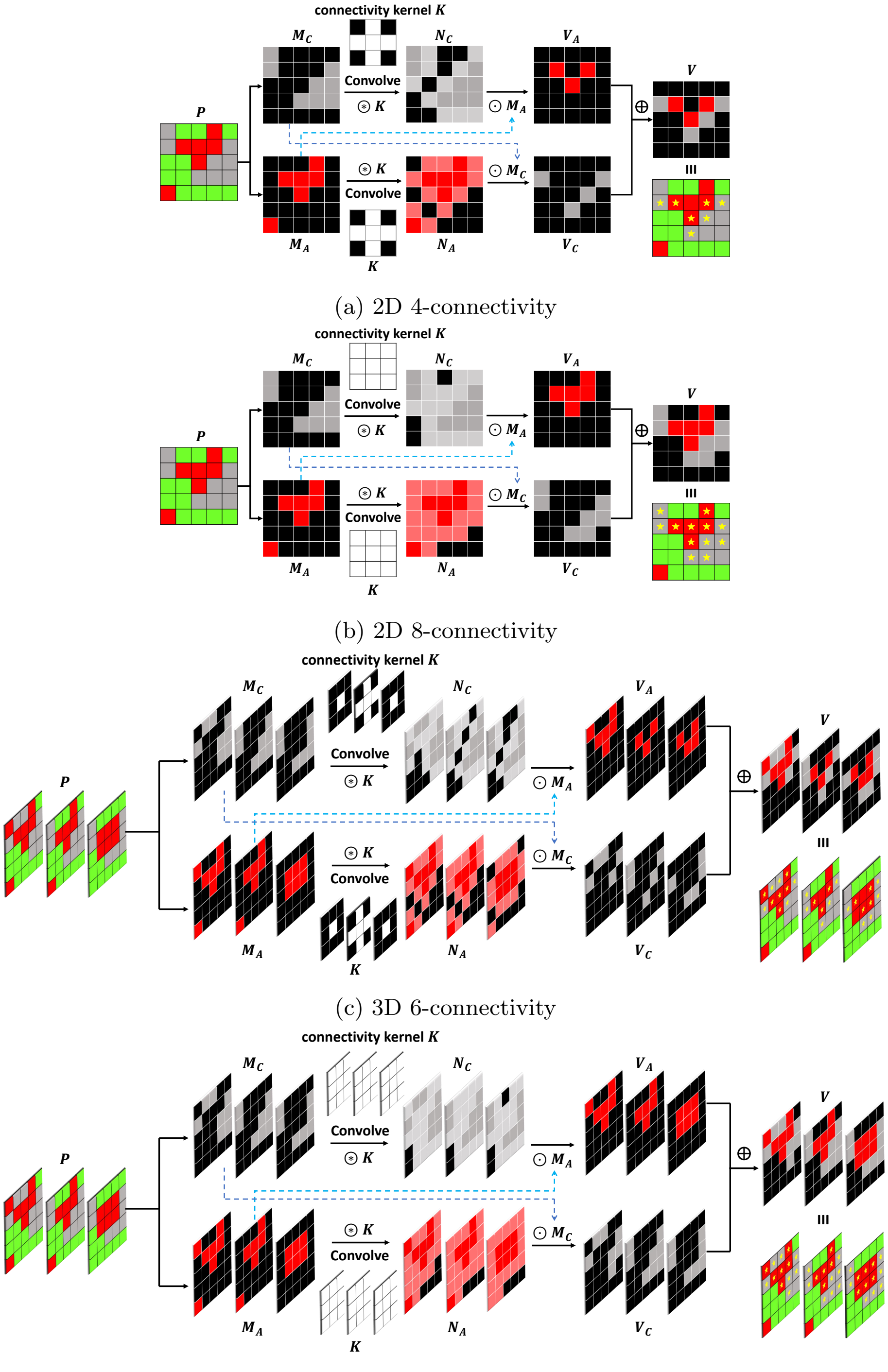
8 Details of the Datasets

Fig. 3 in the main text gives an overview of the classes in each dataset and the topological interactions among them. The datasets are described in more detail as follows.

Aorta. The aorta dataset is a proprietary dataset. 3D CT scans were obtained from 28 randomly selected patients from an institutional database of patients with thoracic and/or abdominal aortic aneurysm. Inclusion criteria for patients included known aneurysmal disease of the aorta and history of undergoing contrast-enhanced CT with arterial-phase contrast injection. Ground truth annotations of the aortic lumen and wall were obtained by four expert readers, working in consensus. Unlike existing aorta datasets[1], our dataset contains accurate aortic wall annotations, which have significant clinical implications. The containment constraint holds as the lumen is completely surrounded by the wall.

IVUS Challenge [1]. The IVUS (IntraVascular UltraSound) challenge is a MICCAI 2011 dataset; we use dataset B in this work. This is a 2D dataset, with each image of dimension 384 × 384. It has been created from in-vivo pullbacks of human coronary arteries and contains lumen and media-adventitia labels. There is a predetermined split of 109 training images and 326 test images. The containment constraints holds as the lumen is completely surrounded by the

[1] https://competitions.codalab.org/competitions/21145



(a) 2D 4-connectivity

(b) 2D 8-connectivity

(c) 3D 6-connectivity

(d) 3D 26-connectivity

Fig. 9: Illustration of the **proposed** strategy to detect the set V of topological critical pixels using different connectivity kernels. The entire critical pixel map V is highlighted with $*$'s.

Table 4: Training Configuration.

Dataset	Model	Patch Size	Batch Size	LR	Optimizer
Aorta	FCN [9]	512 × 512	8	0.01	SGD
	UNet [3]	112 × 112 × 80	2	0.01	momentum 0.99
	nnUNet [6]	160 × 160 × 80	2	0.01	weight decay 3e-5
IVUS	FCN [9]	128 × 128	8	0.01	SGD
	UNet [11]	128 × 128	8	0.01	momentum 0.99
	nnUNet [6]	384 × 384	5	0.01	weight decay 3e-5
Multi-Atlas	FCN [9]	256 × 256	4	0.01	SGD
	UNet [3]	64 × 64 × 32	4	0.01	momentum 0.99
	nnUNet [6]	192 × 192 × 48	2	0.01	weight decay 3e-5
SegTHOR	FCN [9]	256 × 256	4	0.01	SGD
	UNet [3]	64 × 64 × 32	4	0.01	momentum 0.99
	nnUNet [6]	160 × 192 × 64	2	0.01	weight decay 3e-5

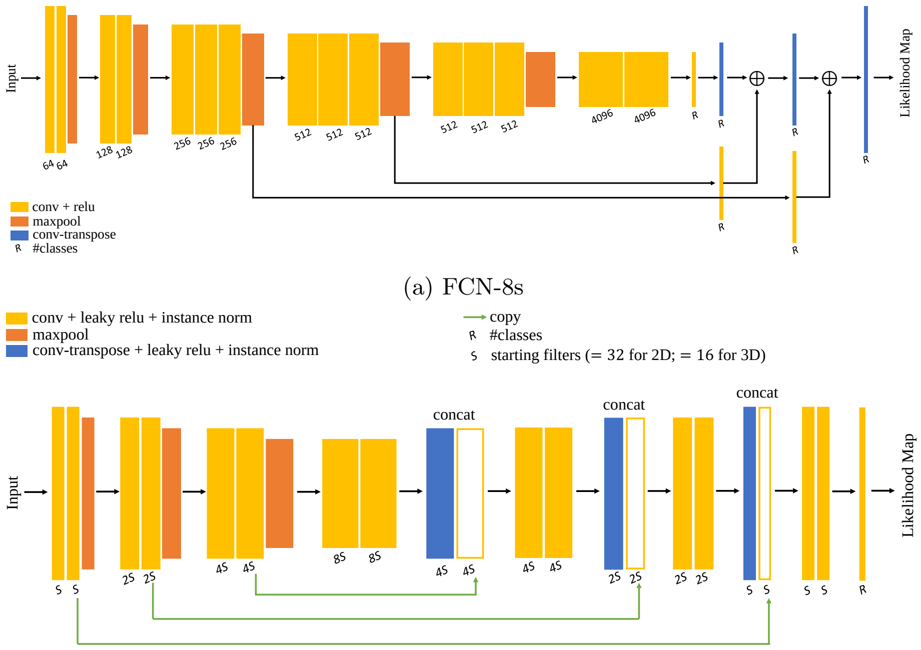
media. The difficulty of this dataset arises due to the imbalanced train-test split, as well as several artifacts (e.g. shadow) in the test set, which causes standard deep neural networks to misclassify the lumen class beyond the media.

Multi-Atlas Labeling Beyond the Cranial Vault [8]. The MICCAI 2015 challenge 'Multi-Atlas Labeling Beyond the Cranial Vault' is a multi-organ segmentation challenge, containing 3D CT scans of the cervix and abdomen. We use the abdomen dataset, which contains thirteen abdominal organ labels. To validate our method, we chose organs that are in close proximity yet exclude each other. We segment four out of the thirteen classes, namely, spleen, left kidney, liver, and stomach. We have clinically verified that the exclusion constraint holds among these four classes, that is, each of these four classes exclude each other. There are 30 volumes available for training, and 20 volumes for testing. The ground truth for the test dataset is available at [5]. We note that while anatomically, the organs follow the exclusion constraint, the available GT did not adhere to it. With the help of clinicians, we have corrected the GT to follow the exclusion constraint. Thus all the baselines were trained on the corrected GT.

SegTHOR [7]. The SegTHOR 2019 challenge dataset contains 3D CT scans of thoracic organs at risk (OAR). In this dataset, the OARs are the heart, the trachea, the aorta and the esophagus, which have varying spatial and appearance characteristics. The dataset contains 40 training volumes and 20 testing volumes. The exclusion constraint holds among three classes, that is, the trachea, the aorta, and the esophagus do not touch each other. We note that while anatomically, the organs follow the exclusion constraint, the available GT did not adhere to it. With the help of clinicians, we have corrected the GT to follow the exclusion constraint. Thus all the baselines were trained on the corrected GT.

9 Implementation Details

We use the PyTorch framework, a single NVIDIA Tesla V100-SXM2 GPU (32G Memory) and a Dual Intel Xeon Silver 4216 CPU@2.1Ghz (16 cores) for all the



(a) FCN-8s

(b) UNet

Fig. 10: Baseline network architectures.

experiments. We use the publicly available codes for UNet [2], FCN [3], nnUNet [4], and NonAdj [5]. The architecture diagrams for the UNet and FCN networks used are shown in Fig. 10. The architecture diagram for nnUNet is not shown as nnUNet uses its planning strategy to generate the best architecture for each dataset.

For the proposed method, the weight term λ_{dice} in the loss function is set to 1.0 by default from nnUNet's planning strategy. We obtain the best results with L_{pixel} set to the cross-entropy loss, $\lambda_{ti} = 1e\text{-}4$ in the 2D setting, and $\lambda_{ti} = 1e\text{-}6$ in the 3D settings.

The training hyperparameters for each network on each dataset is as tabulated in Tab. 4. The loss function used for UNet and FCN is same as that used in vanilla nnUNet, i.e., $L_{ce} + L_{dice}$.

[2] https://github.com/johschmidt42/PyTorch-2D-3D-UNet-Tutorial
[3] https://github.com/pochih/FCN-pytorch
[4] https://github.com/MIC-DKFZ/nnUNet
[5] https://github.com/trypag/NonAdjLoss

Table 5: Ablation study for L_{pixel} (Multi-Atlas)

Class	L_{pixel}	**Dice↑**	**HD↓**	**ASSD↓**	**% Violations↓**
Spleen	None	0.950 ± 0.041	6.084 ± 1.078	0.573 ± 0.131	0.819 ± 0.064
	MSE	0.952 ± 0.025	5.402 ± 1.041	0.492 ± 0.118	0.552 ± 0.071
	DICE	0.957 ± 0.013	5.368 ± 1.042	0.488 ± 0.124	0.493 ± 0.058
	CE	*0.960 ± 0.009*	**5.340 ± 1.049**	**0.484 ± 0.109**	**0.464 ± 0.043**
Kidney	None	0.931 ± 0.018	27.252 ± 5.406	5.352 ± 0.199	/
	MSE	0.934 ± 0.019	22.808 ± 3.186	5.089 ± 0.368	/
	DICE	0.935 ± 0.028	21.935 ± 2.772	4.610 ± 0.465	/
	CE	*0.936 ± 0.026*	**20.013 ± 2.785**	**4.298 ± 0.798**	/
Liver	None	0.951 ± 0.008	38.931 ± 12.161	1.922 ± 0.506	/
	MSE	0.958 ± 0.009	31.672 ± 10.112	1.542 ± 0.628	/
	DICE	0.961 ± 0.009	30.941 ± 9.668	1.195 ± 4.80	/
	CE	**0.962 ± 0.005**	**30.341 ± 9.111**	**0.985 ± 0.386**	/
Stomach	None	0.895 ± 0.015	45.767 ± 7.960	2.720 ± 0.430	/
	MSE	0.905 ± 0.014	39.608 ± 9.717	2.264 ± 0.418	/
	DICE	0.908 ± 0.016	37.763 ± 9.854	1.831 ± 0.402	/
	CE	**0.910 ± 0.018**	**35.514 ± 10.295**	**1.644 ± 0.311**	/

Table 6: Ablation study for λ_{ti} (Multi-Atlas)

Class	λ_{ti}	**Dice↑**	**HD↓**	**ASSD↓**	**% Violations↓**
Spleen	0	0.950 ± 0.041	6.084 ± 1.078	0.573 ± 0.131	0.819 ± 0.064
	5.0e-7	0.954 ± 0.029	5.399 ± 1.034	0.491 ± 0.112	0.541 ± 0.049
	1.0e-6	*0.960 ± 0.009*	**5.340 ± 1.049**	**0.484 ± 0.109**	**0.464 ± 0.043**
	1.5e-6	0.958 ± 0.016	5.361 ± 1.025	0.487 ± 0.122	0.475 ± 0.046
Kidney	0	0.931 ± 0.018	27.252 ± 5.406	5.352 ± 0.199	/
	5.0e-7	0.934 ± 0.022	22.459 ± 3.625	4.936 ± 0.513	/
	1.0e-6	*0.936 ± 0.026*	**20.013 ± 2.785**	**4.298 ± 0.798**	/
	1.5e-6	0.935 ± 0.031	21.360 ± 2.909	4.380 ± 0.687	/
Liver	0	0.951 ± 0.008	38.931 ± 12.161	1.922 ± 0.506	/
	5.0e-7	0.959 ± 0.010	31.390 ± 10.571	1.429 ± 0.421	/
	1.0e-6	**0.962 ± 0.005**	**30.341 ± 9.111**	0.985 ± 0.386	/
	1.5e-6	0.961 ± 0.007	30.586 ± 9.313	**0.966 ± 0.405**	/
Stomach	0	0.895 ± 0.015	45.767 ± 7.960	2.720 ± 0.430	/
	5.0e-7	0.904 ± 0.013	38.984 ± 9.351	2.014 ± 0.477	/
	1.0e-6	**0.910 ± 0.018**	**35.514 ± 10.295**	**1.644 ± 0.311**	/
	1.5e-6	0.908 ± 0.019	36.151 ± 10.192	1.721 ± 0.336	/

10 Additional Ablation Studies

In this section we conduct identical ablation studies as Tab. 3 in the main paper. Here, we conduct this on the Multi-Atlas (exclusion dataset). We report the results in Tab. 5 and Tab. 6. The observation is consistent with the ablation studies on IVUS in the main paper. Using cross-entropy as the surrogate loss function for our topological loss gives the best performance. The method is robust to the choice of the loss weight λ_{ti}. Within a reasonable range, λ_{ti} does impact the performance positively.

11 Additional Results

In all the tables of the main paper, the statistically significant better performances are highlighted with bold. In the supplementary, we highlight in bold the statistically significant better performances within each backbone class (UNet, FCN, nnUNet). The t-test [12] used to determine the statistical significance of the improvement has a confidence interval of 95%. The best, while not statistically significant, performances within each backbone class are highlighted with italics.

We provide comprehensive quantitative results for all the datasets in Tab. 7, 8, 9, and 10. In the tables, 'UNet+Ours' denotes our method trained on the UNet backbone. Similarly, 'FCN+Ours' denotes our method trained on the FCN backbone. We observe that the proposed method improves the quality of segmentations by improving all the metrics significantly compared to the backbone. This supports our claim that our method can be incorporated into any backbone.

We also provide results of our method by changing the connectivity kernel. The default connectivity kernel K, in 2D, is a 3×3 kernel filled with 1's to enforce 8-connectivity. Similarly in 3D, K is a $3 \times 3 \times 3$ kernel filled with 1's to enforce 26-connectivity. For the 2D setting, we also provide results on using the 4-connectivity kernel, which we denote by 'Ours (4conn)' in Tab. 8. For the 3D setting, we also provide results on using the 6-connectivity kernel, which we denote by 'Ours (6conn)' in Tab. 7, 9, and 10. We observe that while using a smaller connectivity kernel does not seem as good as using the default, it is still stronger than other baselines.

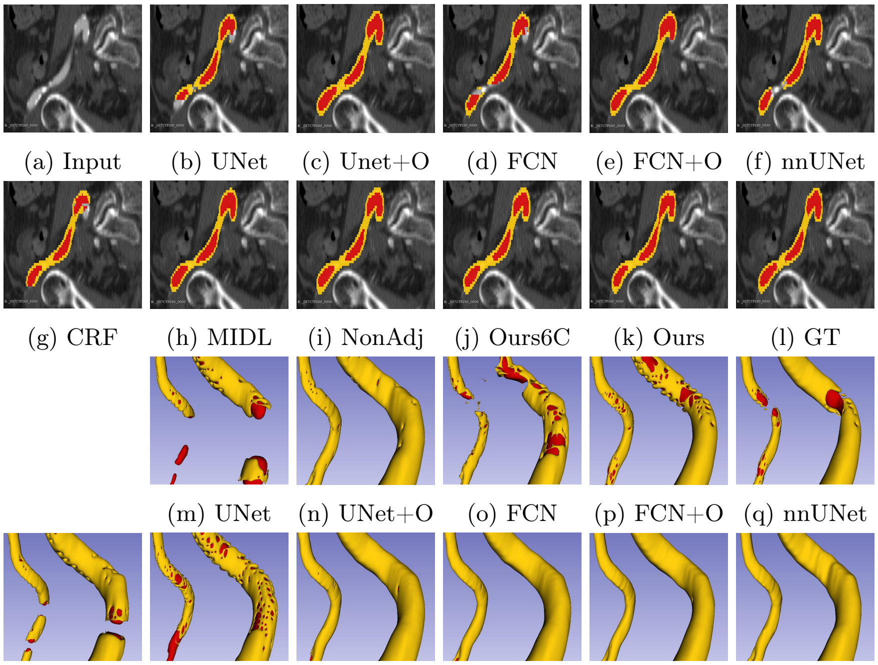
We provide additional qualitative results in Fig. 11, 12, 13, 14, 15, 16, 17, 18, and 19. In the figure sub-captions, 'UNet+O' denotes our method trained on the UNet backbone. Similarly, 'FCN+O' denotes our method trained on the FCN backbone. 'Ours' denotes our method trained on nnUNet with the default connectivity kernel. 'Ours4C' and 'Ours6C' denotes our method trained on nnUNet with the 4-connectivity and 6-connectivity kernel respectively.

Table 7: Quantitative comparison for Aorta dataset (containment constraint)

Class	**Model**	**Dice↑**	**HD↓**	**ASSD↓**	**% Violations↓**
Lumen	UNet [3]	0.900 ± 0.016	64.392 ± 16.874	9.315 ± 1.749	13.994 ± 1.809
	UNet [3] + Ours	**0.918 ± 0.012**	**41.039 ± 10.952**	**6.415 ± 1.403**	**7.734 ± 2.174**
	FCN [9]	*0.894 ± 0.013*	57.974 ± 19.756	9.77 ± 1.421	15.675 ± 2.409
	FCN [9] + Ours	0.892 ± 0.031	**47.772 ± 14.571**	**7.741 ± 1.385**	**9.797 ± 1.707**
	nnUNet [6]	0.906 ± 0.020	36.368 ± 12.559	4.563 ± 0.675	5.424 ± 2.461
	Topo-CRF [2]	0.897 ± 0.057	40.162 ± 18.687	5.952 ± 0.999	8.358 ± 2.151
	MIDL [10]	0.912 ± 0.008	32.157 ± 16.270	6.405 ± 0.524	6.377 ± 1.661
	NonAdj [4]	0.916 ± 0.030	32.465 ± 18.848	4.771 ± 1.129	4.932 ± 1.479
	Ours (6conn)	0.920 ± 0.006	29.693 ± 15.746	4.269 ± 0.995	3.706 ± 1.274
	Ours (26conn)	**0.922 ± 0.009**	**25.959 ± 13.574**	**3.920 ± 0.765**	**3.526 ± 1.244**
Wall	UNet [3]	0.677 ± 0.015	71.109 ± 24.653	12.497 ± 1.372	/
	UNet [3] + Ours	**0.737 ± 0.024**	**44.372 ± 11.702**	**7.289 ± 0.792**	/
	FCN [9]	0.651 ± 0.015	66.059 ± 17.188	12.339 ± 0.959	/
	FCN [9] + Ours	**0.681 ± 0.023**	**50.068 ± 4.469**	**9.530 ± 1.275**	/
	nnUNet [6]	0.741 ± 0.026	42.486 ± 15.139	8.005 ± 0.811	/
	Topo-CRF [2]	0.739 ± 0.010	46.873 ± 17.636	7.914 ± 0.877	/
	MIDL [10]	0.742 ± 0.028	43.132 ± 15.624	6.420 ± 1.242	/
	NonAdj [4]	0.748 ± 0.017	38.197 ± 19.598	4.887 ± 0.702	/
	Ours (6conn)	0.753 ± 0.015	35.977 ± 17.358	4.200 ± 0.738	/
	Ours (26conn)	**0.758 ± 0.017**	**31.137 ± 17.772**	**5.799 ± 0.737**	/

Table 8: Quantitative comparison for IVUS dataset (containment constraint)

Class	Model	Dice↑	HD↓	ASSD↓	% Violations↓
Lumen	UNet [11]	0.786 ± 0.144	6.643 ± 1.936	30.944 ± 11.631	5.970 ± 2.141
	UNet [11] + Ours	**0.843 ± 0.128**	**4.258 ± 1.612**	**21.597 ± 9.138**	**2.042 ± 1.320**
	FCN [9]	0.824 ± 0.071	5.319 ± 1.519	22.551 ± 7.882	3.766 ± 1.444
	FCN [9] + Ours	**0.871 ± 0.082**	**3.976 ± 1.207**	**11.531 ± 4.736**	**1.752 ± 1.105**
	nnUNet [6]	0.893 ± 0.066	3.464 ± 0.917	11.152 ± 3.954	2.708 ± 1.032
	Topo-CRF [2]	0.887 ± 0.096	4.138 ± 1.454	10.497 ± 2.487	2.371 ± 0.960
	MIDL [10]	0.891 ± 0.073	4.226 ± 1.390	10.641 ± 2.322	2.394 ± 0.918
	NonAdj [4]	0.897 ± 0.081	3.140 ± 1.154	9.628 ± 3.221	2.173 ± 0.994
	Ours (4conn)	0.912 ± 0.087	2.857 ± 0.949	6.710 ± 3.186	0.311 ± 0.927
	Ours (8conn)	**0.949 ± 0.070**	**2.046 ± 1.079**	**6.057 ± 2.746**	**0.157 ± 0.808**
Media	UNet [11]	0.651 ± 0.130	7.391 ± 1.072	21.984 ± 6.634	/
	UNet [11] + Ours	**0.688 ± 0.115**	**7.012 ± 0.983**	**18.651 ± 5.776**	/
	FCN [9]	0.782 ± 0.144	6.806 ± 1.147	13.863 ± 4.511	/
	FCN [9] + Ours	**0.809 ± 0.127**	**6.137 ± 1.093**	**9.115 ± 3.689**	/
	nnUNet [6]	0.856 ± 0.090	5.646 ± 1.228	6.491 ± 2.314	/
	Topo-CRF [2]	0.843 ± 0.106	5.409 ± 1.166	5.929 ± 1.785	/
	MIDL [10]	0.841 ± 0.121	5.461 ± 1.214	6.071 ± 1.837	/
	NonAdj [4]	0.848± 0.117	5.983 ± 1.342	6.615 ± 1.937	/
	Ours (4conn)	0.884 ± 0.094	4.188 ± 1.156	3.622 ± 2.008	/
	Ours (8conn)	**0.910 ± 0.089**	**3.873 ± 0.933**	**3.171 ± 1.871**	/



(a) Input (b) UNet (c) Unet+O (d) FCN (e) FCN+O (f) nnUNet

(g) CRF (h) MIDL (i) NonAdj (j) Ours6C (k) Ours (l) GT

(m) UNet (n) UNet+O (o) FCN (p) FCN+O (q) nnUNet

(r) CRF (s) MIDL (t) NonAdj (u) Ours6C (v) Ours (w) GT

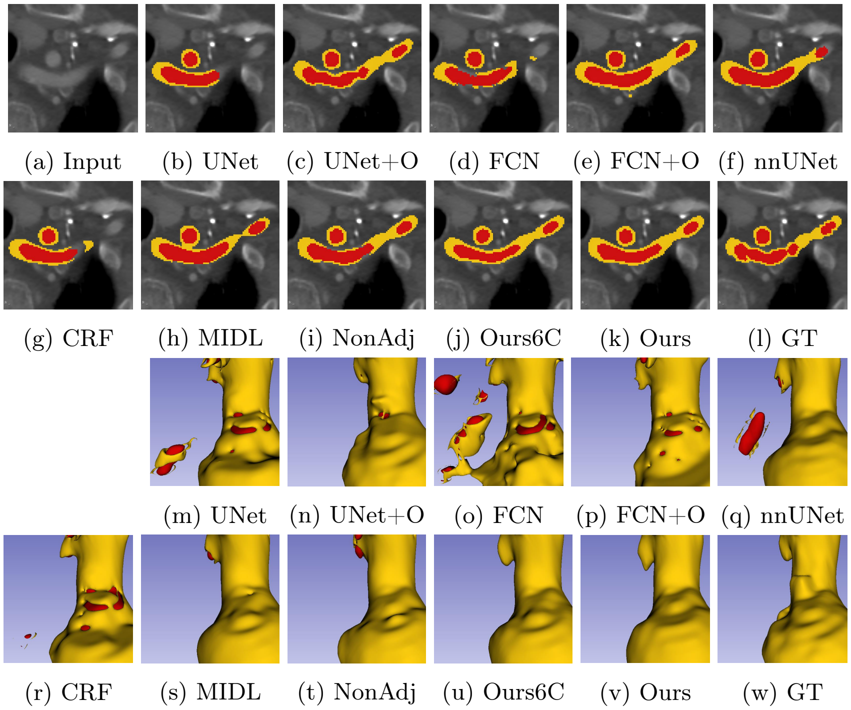
Fig. 11: Qualitative Aorta results compared with the baselines. Rows 3-4 are corresponding 3D renderings. It is hard to visualize the input 3D volumetric image and so we leave it blank in the third row. Colors for the classes correspond to the ones used in Fig. 3.

Table 9: Quantitative comparison for Multi-Atlas dataset (exclusion constraint)

Class	**Model**	**Dice↑**	**HD↓**	**ASSD↓**	**% Violations↓**
Spleen	UNet [3]	0.919 ± 0.041	47.037 ± 17.365	4.323 ± 0.367	1.857 ± 0.123
	UNet [3] + Ours	*0.932 ± 0.059*	**34.445 ± 10.684**	**2.020 ± 0.218**	**1.256 ± 0.153**
	FCN [9]	0.909 ± 0.037	134.915 ± 65.623	17.646 ± 10.604	3.041 ± 0.181
	FCN [9] + Ours	**0.927 ± 0.011**	**66.407 ± 9.946**	**9.038 ± 2.146**	**2.680 ± 0.128**
	nnUNet [6]	0.950 ± 0.041	6.084 ± 1.078	0.573 ± 0.131	0.819 ± 0.064
	Topo-CRF [2]	0.947 ± 0.028	6.403 ± 1.039	1.844 ± 0.517	0.934 ± 0.032
	MIDL [10]	0.944 ± 0.015	5.597 ± 1.374	0.565 ± 0.124	0.725 ± 0.151
	NonAdj [4]	0.952 ± 0.058	5.621 ± 1.065	0.513 ± 0.175	0.521 ± 0.082
	Ours (6conn)	0.957 ± 0.023	5.395 ± 1.057	0.498 ± 0.127	0.486 ± 0.075
	Ours (26conn)	*0.960 ± 0.009*	**5.340 ± 1.049**	**0.484 ± 0.109**	**0.464 ± 0.043**
Kidney	UNet [3]	0.908 ± 0.079	61.602 ± 13.168	9.992 ± 2.461	/
	UNet [3] + Ours	*0.921 ± 0.023*	**42.525 ± 10.103**	**6.446 ± 1.404**	/
	FCN [9]	0.892 ± 0.018	187.472 ± 36.096	11.583 ± 2.396	/
	FCN [9] + Ours	**0.916 ± 0.014**	**93.283 ± 10.293**	**8.675 ± 1.129**	/
	nnUNet [6]	0.931 ± 0.018	27.252 ± 5.406	5.352 ± 0.199	/
	Topo-CRF [2]	0.928 ± 0.059	30.209 ± 5.317	6.308 ± 0.905	/
	MIDL [10]	0.935 ± 0.071	25.208 ± 5.440	4.885 ± 0.421	/
	NonAdj [4]	0.934 ± 0.012	24.182 ± 5.561	4.692 ± 0.657	/
	Ours (6conn)	0.932 ± 0.013	23.176 ± 3.593	4.540 ± 0.883	/
	Ours (26conn)	*0.936 ± 0.026*	**20.013 ± 2.785**	**4.298 ± 0.798**	/
Liver	UNet [3]	0.912 ± 0.016	64.556 ± 13.894	2.324 ± 0.513	/
	UNet [3] + Ours	**0.941 ± 0.038**	**46.174 ± 11.744**	**1.452 ± 0.717**	/
	FCN [9]	0.885 ± 0.034	183.870 ± 49.796	29.061 ± 13.484	/
	FCN [9] + Ours	**0.937 ± 0.013**	**117.200 ± 16.663**	**7.324 ± 5.201**	/
	nnUNet [6]	0.951 ± 0.008	38.931 ± 12.161	1.922 ± 0.506	/
	Topo-CRF [2]	0.949 ± 0.006	46.449 ± 14.188	2.072 ± 0.313	/
	MIDL [10]	0.955 ± 0.005	34.276 ± 11.253	1.344 ± 0.431	/
	NonAdj [4]	0.957 ± 0.003	33.671 ± 13.543	1.185 ± 0.372	/
	Ours (6conn)	0.958 ± 0.006	32.674 ± 12.566	1.098 ± 0.405	/
	Ours (26conn)	**0.962 ± 0.005**	**30.341 ± 9.111**	**0.985 ± 0.386**	/
Stomach	UNet [3]	0.846 ± 0.084	76.000 ± 24.352	5.023 ± 1.508	/
	UNet [3] + Ours	*0.872 ± 0.074*	**54.039 ± 19.131**	**3.611 ± 1.301**	/
	FCN [9]	0.708 ± 0.156	172.855 ± 43.735	11.328 ± 3.178	/
	FCN [9] + Ours	*0.799 ± 0.127*	**104.331 ± 10.276**	**6.892 ± 1.905**	/
	nnUNet [6]	0.895 ± 0.015	45.767 ± 7.960	2.720 ± 0.430	/
	Topo-CRF [2]	0.888 ± 0.015	46.877 ± 9.861	3.675 ± 0.358	/
	MIDL [10]	0.899 ± 0.012	40.282 ± 6.437	2.567 ± 0.431	/
	NonAdj [4]	0.907 ± 0.028	41.749 ± 8.630	2.184 ± 0.325	/
	Ours (6conn)	0.908 ± 0.017	39.853 ± 9.544	1.879 ± 0.587	/
	Ours (26conn)	**0.910 ± 0.018**	**35.514 ± 10.295**	**1.644 ± 0.311**	/

Table 10: Quantitative comparison for SegTHOR dataset (exclusion constraint)

Class	**Model**	**Dice↑**	**HD↓**	**ASSD↓**	**% Violations↓**
Esophagus	UNet [3]	0.827 ± 0.038	11.357 ± 2.709	1.186 ± 0.113	3.212 ± 0.720
	UNet [3] + Ours	*0.841 ± 0.026*	**8.916 ± 2.437**	**0.970 ± 0.124**	**2.559 ± 0.412**
	FCN [9]	0.800 ± 0.031	10.770 ± 2.085	1.303 ± 0.128	3.616 ± 0.709
	FCN [9] + Ours	**0.839 ± 0.027**	**9.055 ± 2.681**	**0.986 ± 0.108**	**2.889 ± 0.618**
	nnUNet [6]	0.841 ± 0.014	8.018 ± 2.085	0.950 ± 0.070	1.947 ± 0.525
	Topo-CRF [2]	0.839 ± 0.029	8.602 ± 2.363	0.991 ± 0.081	2.070 ± 0.687
	MIDL [10]	0.840 ± 0.020	7.266 ± 2.132	0.921 ± 0.136	1.271 ± 0.912
	NonAdj [4]	0.843 ± 0.020	6.293 ± 2.703	0.897 ± 0.078	1.215 ± 0.211
	Ours (6conn)	0.849 ± 0.014	5.774 ± 2.371	0.832 ± 0.074	0.911 ± 0.565
	Ours (26conn)	**0.858 ± 0.019**	**5.582 ± 2.250**	**0.798 ± 0.042**	**0.749 ± 0.428**
Trachea	UNet [3]	0.897 ± 0.027	10.656 ± 4.047	0.728 ± 0.146	/
	UNet [3] + Ours	*0.908 ± 0.041*	*8.957 ± 3.338*	**0.592 ± 0.167**	/
	FCN [9]	0.891 ± 0.031	11.789 ± 5.291	0.953 ± 0.221	/
	FCN [9] + Ours	*0.896 ± 0.035*	*9.620 ± 2.805*	**0.683 ± 0.245**	/
	nnUNet [6]	0.910 ± 0.018	9.423 ± 2.393	0.478 ± 0.152	/
	Topo-CRF [2]	0.909 ± 0.022	10.435 ± 2.334	0.473 ± 0.167	/
	MIDL [10]	0.914 ± 0.027	7.929 ± 2.305	0.456 ± 0.144	/
	NonAdj [4]	0.913 ± 0.028	7.866 ± 2.343	0.440 ± 0.113	/
	Ours (6conn)	0.922 ± 0.031	7.851 ± 2.846	0.417 ± 0.157	/
	Ours (26conn)	**0.929 ± 0.020**	**7.280 ± 2.109**	**0.316 ± 0.186**	/
Aorta	UNet [3]	0.929 ± 0.020	9.716 ± 4.032	0.714 ± 0.293	/
	UNet [3] + Ours	*0.932 ± 0.029*	**6.553 ± 3.932**	*0.697 ± 0.218*	/
	FCN [9]	0.924 ± 0.021	9.869 ± 4.739	0.726 ± 0.424	/
	FCN [9] + Ours	*0.929 ± 0.025*	**6.751 ± 3.810**	*0.705 ± 0.263*	/
	nnUNet [6]	0.935 ± 0.017	5.353 ± 2.698	0.658 ± 0.177	/
	Topo-CRF [2]	0.932 ± 0.018	5.361 ± 2.763	0.690 ± 0.225	/
	MIDL [10]	0.937 ± 0.016	5.349 ± 2.458	0.668 ± 0.128	/
	NonAdj [4]	0.939 ± 0.021	5.060 ± 2.345	0.638 ± 0.192	/
	Ours (6conn)	0.940 ± 0.017	4.840 ± 2.859	0.621 ± 0.175	/
	Ours (26conn)	*0.942 ± 0.018*	*4.758 ± 2.127*	*0.606 ± 0.214*	/
Heart	UNet [3]	0.948 ± 0.012	8.235 ± 4.382	1.158 ± 0.571	/
	UNet [3] + Ours	*0.953 ± 0.013*	*7.454 ± 4.602*	*1.022 ± 0.633*	/
	FCN [9]	0.948 ± 0.014	8.556 ± 4.302	2.206 ± 0.905	/
	FCN [9] + Ours	*0.950 ± 0.018*	*8.085 ± 4.637*	**1.543 ± 0.596**	/
	nnUNet [6]	0.956 ± 0.014	7.732 ± 4.327	0.895 ± 0.328	/
	Topo-CRF [2]	0.954 ± 0.016	7.936 ± 4.665	1.022 ± 0.434	/
	MIDL [10]	0.952 ± 0.014	7.615 ± 4.991	0.889 ± 0.371	/
	NonAdj [4]	0.956 ± 0.016	7.363 ± 4.609	0.895 ± 0.382	/
	Ours (6conn)	0.958 ± 0.013	7.316 ± 4.129	0.874 ± 0.372	/
	Ours (26conn)	*0.959 ± 0.012*	*7.158 ± 4.355*	*0.871 ± 0.363*	/

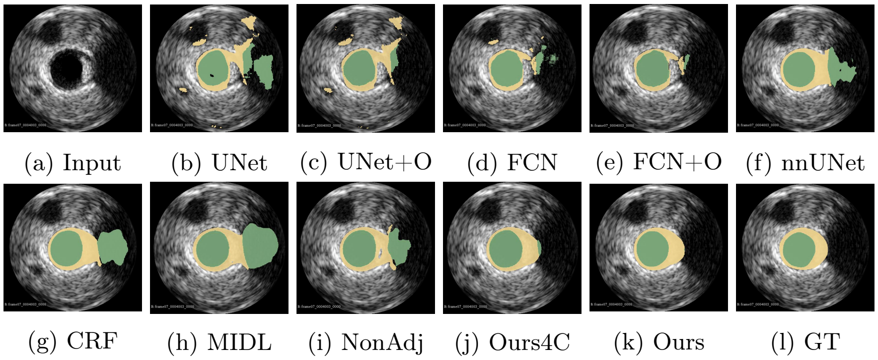

(a) Input (b) UNet (c) UNet+O (d) FCN (e) FCN+O (f) nnUNet

(g) CRF (h) MIDL (i) NonAdj (j) Ours6C (k) Ours (l) GT

(m) UNet (n) UNet+O (o) FCN (p) FCN+O (q) nnUNet

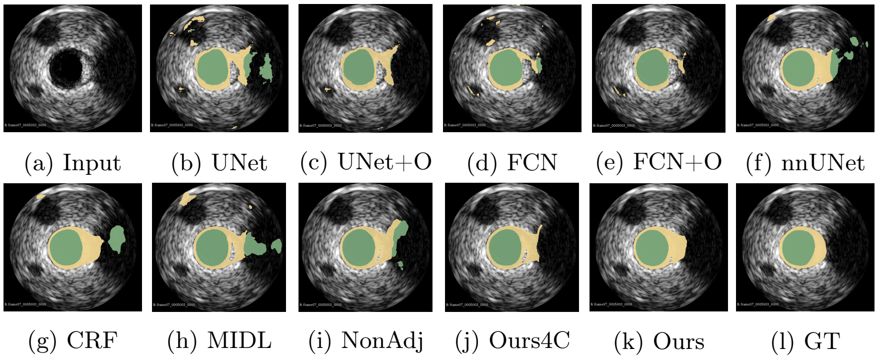
(r) CRF (s) MIDL (t) NonAdj (u) Ours6C (v) Ours (w) GT

Fig. 12: Additional qualitative Aorta results compared with the baselines. Rows 3-4 are corresponding 3D renderings. It is hard to visualize the input 3D volumetric image and so we leave it blank in the third row. Colors for the classes correspond to the ones used in Fig. 3.


(a) Input (b) UNet (c) UNet+O (d) FCN (e) FCN+O (f) nnUNet

(g) CRF (h) MIDL (i) NonAdj (j) Ours4C (k) Ours (l) GT

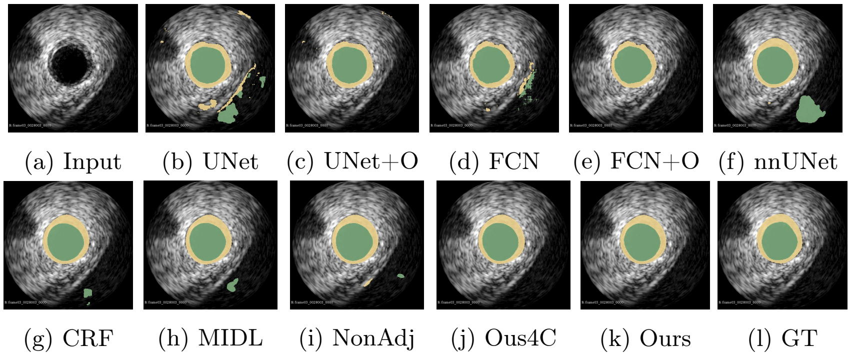
Fig. 13: Qualitative IVUS results compared with the baselines. Colors for the classes correspond to the ones used in Fig. 3.



(a) Input (b) UNet (c) UNet+O (d) FCN (e) FCN+O (f) nnUNet

(g) CRF (h) MIDL (i) NonAdj (j) Ours4C (k) Ours (l) GT

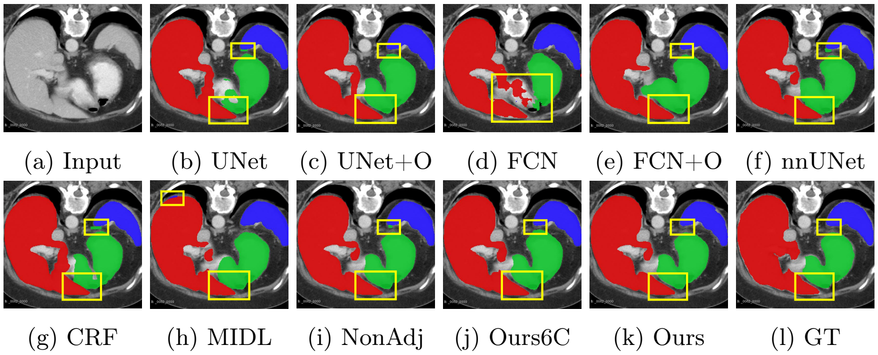
Fig. 14: Additional qualitative IVUS results compared with the baselines. Colors for the classes correspond to the ones used in Fig. 3.



(a) Input (b) UNet (c) UNet+O (d) FCN (e) FCN+O (f) nnUNet

(g) CRF (h) MIDL (i) NonAdj (j) Ous4C (k) Ours (l) GT

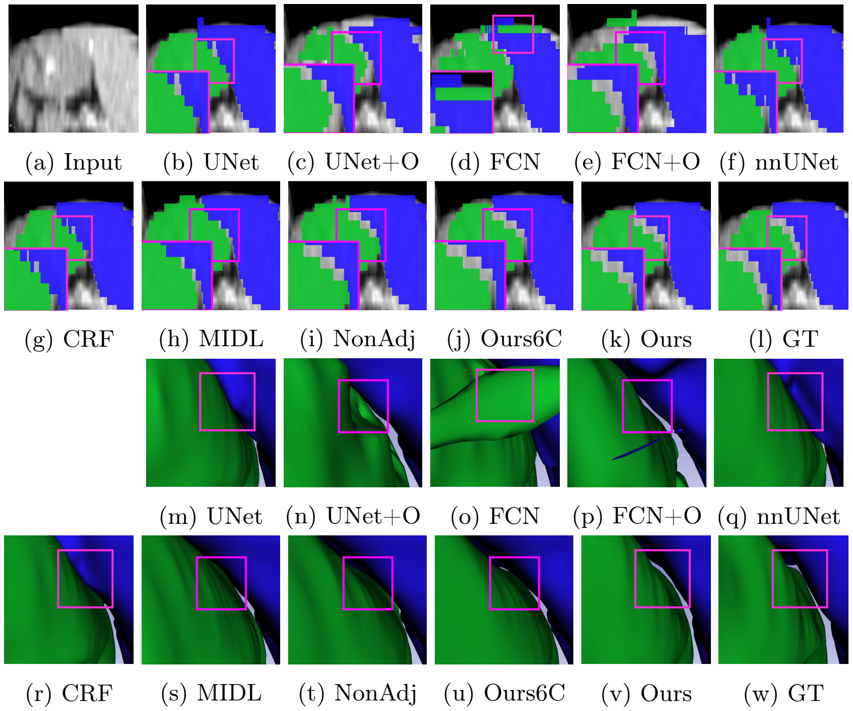
Fig. 15: Additional qualitative IVUS results compared with the baselines. Colors for the classes correspond to the ones used in Fig. 3.



(a) Input (b) UNet (c) UNet+O (d) FCN (e) FCN+O (f) nnUNet

(g) CRF (h) MIDL (i) NonAdj (j) Ours6C (k) Ours (l) GT

Fig. 16: Qualitative Multi-Atlas results compared with the baselines. Colors for the classes correspond to the ones used in Fig. 3.



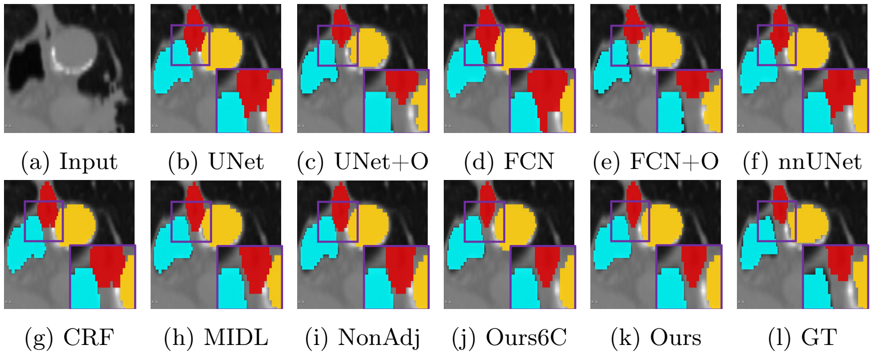
(a) Input (b) UNet (c) UNet+O (d) FCN (e) FCN+O (f) nnUNet

(g) CRF (h) MIDL (i) NonAdj (j) Ours6C (k) Ours (l) GT

(m) UNet (n) UNet+O (o) FCN (p) FCN+O (q) nnUNet

(r) CRF (s) MIDL (t) NonAdj (u) Ours6C (v) Ours (w) GT

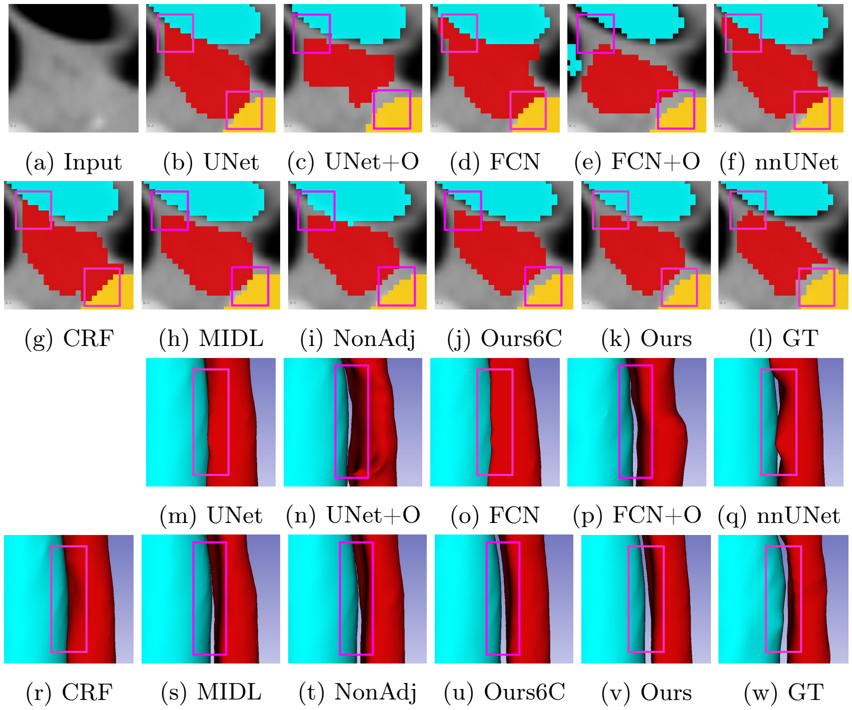
Fig. 17: Additional qualitative Multi-Atlas results compared with the baselines. Rows 3-4 are corresponding 3D renderings. It is hard to visualize the input 3D volumetric image and so we leave it blank in the third row. Colors for the classes correspond to the ones used in Fig. 3.



(a) Input (b) UNet (c) UNet+O (d) FCN (e) FCN+O (f) nnUNet

(g) CRF (h) MIDL (i) NonAdj (j) Ours6C (k) Ours (l) GT

Fig. 18: Qualitative SegTHOR results compared with the baselines. Colors for the classes correspond to the ones used in Fig. 3.



(a) Input (b) UNet (c) UNet+O (d) FCN (e) FCN+O (f) nnUNet

(g) CRF (h) MIDL (i) NonAdj (j) Ours6C (k) Ours (l) GT

(m) UNet (n) UNet+O (o) FCN (p) FCN+O (q) nnUNet

(r) CRF (s) MIDL (t) NonAdj (u) Ours6C (v) Ours (w) GT

Fig. 19: Additional qualitative SegTHOR results compared with the baselines. Rows 3-4 are corresponding 3D renderings. It is hard to visualize the input 3D volumetric image and so we leave it blank in the third row. Colors for the classes correspond to the ones used in Fig. 3.

References

1. Balocco, S., Gatta, C., Ciompi, F., Wahle, A., Radeva, P., Carlier, S., Unal, G., Sanidas, E., Mauri, J., Carillo, X., et al.: Standardized evaluation methodology and reference database for evaluating ivus image segmentation. Computerized medical imaging and graphics **38**(2), 70–90 (2014)
2. BenTaieb, A., Hamarneh, G.: Topology aware fully convolutional networks for histology gland segmentation. In: International conference on medical image computing and computer-assisted intervention. pp. 460–468. Springer (2016)
3. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: MICCAI (2016)
4. Ganaye, P.A., Sdika, M., Triggs, B., Benoit-Cattin, H.: Removing segmentation inconsistencies with semi-supervised non-adjacency constraint. Medical image analysis **58**, 101551 (2019)
5. Gibson, E., Giganti, F., Hu, Y., Bonmati, E., Bandula, S., Gurusamy, K., Davidson, B., Pereira, S.P., Clarkson, M.J., Barratt, D.C.: Multi-organ abdominal ct reference standard segmentations (feb 2018). https://doi.org/10.5281/zenodo.1169361
6. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods (2021)
7. Lambert, Z., Petitjean, C., Dubray, B., Ruan, S.: Segthor: Segmentation of thoracic organs at risk in ct images (2019)
8. Landman, B., Xu, Z., Igelsias, J., Styner, M., Langerak, T., Klein, A.: Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge. In: Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge. vol. 5, p. 12 (2015)
9. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
10. Reddy, C., Gopinath, K., Lombaert, H.: Brain tumor segmentation using topological loss in convolutional networks. In: International Conference on Medical Imaging with Deep Learning–Extended Abstract Track (2019)
11. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
12. Student: The probable error of a mean. Biometrika pp. 1–25 (1908)