Deep learning for vascular segmentation and tissue characterization in CT images

Zhang, X.

Citation

Zhang, X. (2026, January 7). *Deep learning for vascular segmentation and tissue characterization in CT images*. Retrieved from https://hdl.handle.net/1887/4286096

5

Point-based Couinaud segmentation without liver vessel priors

This chapter was adapted from:

Zhang, X., Broersen, A., van Erp, G., Pintea, S.L. and Dijkstra, J., **Skip priors and add graph-based anatomical information, for point-based Couinaud segmentation.** (2025) In Reconstruction and Imaging Motion Estimation, and Graphs in Biomedical Image Analysis: First International Workshop, RIME 2025, and 7th International Workshop, GRAIL 2025, Daejeon, South Korea, September 27, 2025, Proceedings. Springer-Verlag, Berlin, Heidelberg, 131-140.

Abstract

The preoperative planning of liver surgery relies on Couinaud segmentation to reduce the risk of bleeding and guide the resection procedure. Using 3D point-based representations, rather than voxelizing the CT volume, has the benefit of preserving the physical resolution of the CT. However, point-based representations need prior knowledge of the liver vessel structure, which is time consuming to acquire. Here, we propose a point-based method for Couinaud segmentation, without explicitly providing the liver vessel structure. To allow the model to learn the anatomical structure, without providing the liver vessel structure, we add a graph reasoning module on top of the point features. This adds implicit contextual information to the model, by learning affinities across CT-voxels. Our method has competitive results on the *MSD* and *LiTS* public datasets in Dice coefficient and average surface distance scores compared to four state-of-the-art point-based baselines.

5.1 Introduction

Primary liver cancer, the sixth most common cancer worldwide, is predominantly caused by HCC (Hepatocellular Carcinoma) [1]. Effective treatment for primary liver cancer relies on two main procedures: liver resection and radiofrequency ablation [1]. Both approaches depend on accurate Couinaud segmentation, to reduce the risk of main vessel puncture, and to guide the placement of ablation needles [2, 3]. Couinaud segmentation divides the liver into eight functionally independent segments. The right-, middle- and left-hepatic veins divide the liver into four sections. These sections are then further split by the horizontal plane defined by the portal vein. In clinical practice, as shown in Fig. 5.1, Couinaud segments are manually delineated based on the hepatic and portal veins.

Prior work for automatically Couinaud segmentation, voxelizes the liver CT to be used in 3D convolutional neural networks (CNNs) [4, 5, 6, 7]. More recently [8], computes point embeddings from sampled 3D points within the liver area. Using sample 3D points has the added value that they preserve the physical CT resolution without the need to resize or crop along the axial direction. Therefore, here we restrict our focus to point-based methods. While relying on 3D point-based representations, our proposed method does not need prior liver vessel information, unlike Zhang et al. [8]. Yet, without prior liver vessel knowledge, we lose anatomical information. To incorporate the anatomic structure, we add a graph reasoning module that learns affinities between the embeddings of the 3D points.

To summarize: (i) We propose a 3D point-based method for Couinaud segmentation that does not rely on anatomic prior knowledge of the liver vessel structure; (ii) We use graph reasoning to learn affinities between different points along the liver and, thus, implicitly learn anatomic structure; (iii) We evaluate our method on two public datasets: *MSD* and *LiTS* and show competitive accuracy when compared to four state-of-the art point-based segmentation methods.

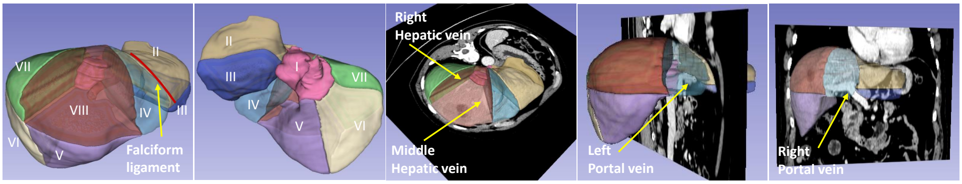

Figure 5.1: Couinaud segmentation is challenging because it requires prior knowledge of the liver vessels.

5.2 Related work

5.2.1 Couinaud segmentation

Prior work on automatic Couinaud segmentation creates liver atlases [9, 10], divides the liver into voxels to be used in 3D convolutional neural networks (CNNs) [4, 5, 6, 7], or builds deep models on top of sampled 3D points [8, 11]. Atlas-based and partial CNN-based methods [4, 9, 10, 7] require manual landmarks along the hepatic veins, whereas the other prior-free CNN-based methods [5, 6] on CT images need to resize the CT volume to a fixed grid size which changes the physical resolution of the CT images. Point-based models [8, 11] address the limitation of voxelized methods, while still requiring prior liver vessel information. Here, we build on 3D point-based methods, while discarding the need for prior anatomical information, and learning this implicitly via dynamic graph reasoning.

5.2.2 Dynamic graph reasoning

Dynamic graph reasoning is widely used in both image-based [12, 13, 14, 15] as well as point-based semantic segmentation methods [16, 17, 18], to capture long-range dependencies. Most image-based methods [12, 13, 14] consider all position pairs when calculating affinities, resulting in high complexity. Unlike these methods, DGMN [15] proposed an adaptive sampling method that considers only limited positions. Similarly, point-based methods also suffer from high complexity of affinity calculation. K-nearest neighbor (k-NN) is typically used for the complexity reduction, as in [17, 18]. Alternatively, Ma et al. [16] propose to learn channel dependencies instead of dependencies between nodes to capture global contextual information while reducing the computations. Here, we take advantage of both k-NN in the point domain, and the adaptive sampling method in [15] to reduce computations.

5.3 Methods

Our model starts from a set of 3D points, $\mathbf{p}$, sampled from the liver region over the complete CT volume, and their associated intensities. We follow the design of adaptive graph CNN (AGCNN) [19], processing the points at four levels, as in Fig. 5.2. We extend AGCNN with the green blocks: the grid feature embeddings, $f(\mathbf{p})$, enhancing the contextual information; and graph reasoning model, $\mathcal{G}r(f(\mathbf{p}))$, which dynamically learns affinities across points. At the last level, we interpolate the point embeddings, $\mathbf{p}_4$, and feed the result to an MLP with two layers ($\{64, 8\}$). We predict the eight Couinaud segments in the liver region of the CT volume. We use the cross-entropy loss to train the model.

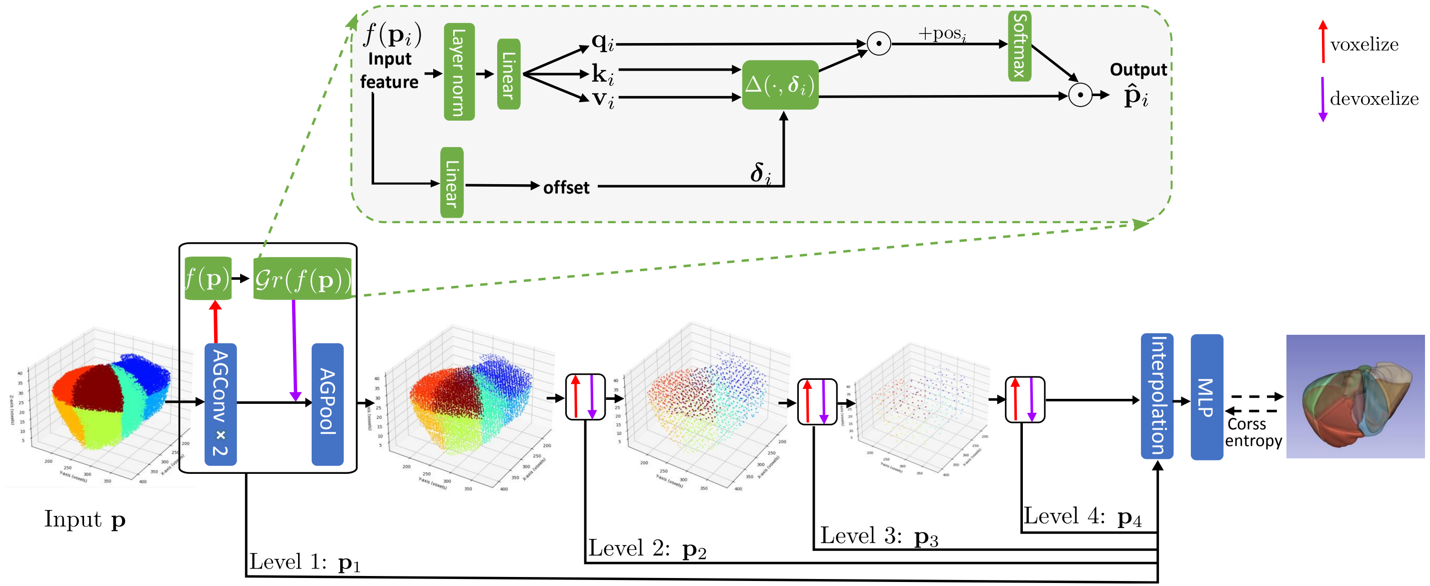

Figure 5.2: Network architecture. We build on the design of [19], and extend this with the green blocks: grid feature embeddings, $f(\mathbf{p})$ adding contextual information; and graph reasoning model, $\mathcal{G}r(f(\mathbf{p}))$, learning dynamic affinities in neighborhood areas.

5.3.1 Relation to adaptive graph CNN

AGCNN starts from a set of 3D points, $\mathbf{p}$, and their associated CT intensities. Before training, AGCNN precomputes a set of $K \ll N$ neighbors for each point, using a ball-query sampling scheme, with radius r, as in Fig. 5.3. At the first level the neighbors of point j, $\mathbf{p}^{ne(j)}$, are taken from the initial set of points $\mathbf{p}$, while at lower resolution levels, the neighbors are taken from the previous level points, $\mathbf{p}_{i-1}, i \in \{2,3,4\}$.

To add implicit anatomical vessel structure to AGCNN, we use a graph reasoning module $\mathcal{G}$. Specifically, at each level i we voxelize all N_i points, $\mathbf{p_i}$ into a grid of size $[M_i \times M_i \times M_i]$, using the method of Liu *et al.* [20]. In this voxelized space, the graph reasoning module $\mathcal{G}r(\cdot)$ dynamically learns which voxels are informative. And it learns to select 3^3 voxels with which to compute affinities. Intuitively, AGCNN computes affinities of points in the current level with the previous level, while our graph reasoning module computes affinities among the current level points.

5.3.2 Implicit anatomic contextual knowledge

Grid feature embeddings. The inputs to the model are a set of 3D points in $\mathbb{R}^3$, together with their corresponding CT intensity values. We revoxelize the point features and obtain the voxelized point embeddings, $f(\mathbf{p}_i)$ at each level, i, by performing two 3D residual convolutions.

Dynamic graph reasoning. The grid-space $[M_i \times M_i \times M_i]$ at each level i contains many voxels $\left(\geq \left(\frac{32}{2^{i-1}}\right)^3\right)$ at different levels. This causes memory bottlenecks when calculating the affinity matrix. To address this, we draw inspiration from DGMN [15] and extend their method from 2D to 3D. Specifically, we want to consider only a subset of 3^3 voxels out of the $[M_i \times M_i \times M_i]$ grid, when computing affinities. Following DGMN [15], we use self-attention [21] to define affinities between voxelized points. Given the voxelized point feature $f(\mathbf{p}_i)$ at level i, we first project this to *query* $\mathbf{q}_i$, *key* $\mathbf{k}_i$,

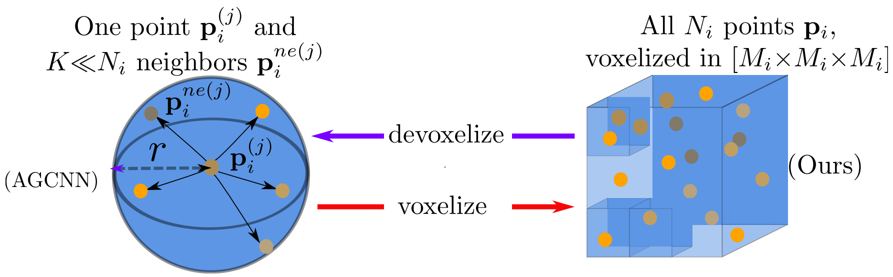

Figure 5.3: **Left:** AGCNN computes offline a set of $K \ll N$ neighbors $\mathbf{p}^{ne(j)}$ for each point $\mathbf{p}^{(j)}$ to be used in the graph affinities. **Right:** Our approach voxelizes all N points, $\mathbf{p}$, into a grid $[M \times M \times M]$, and dynamically focuses on 3^3 voxels when learning affinities.

and *value* $\mathbf{v}_i$, via a shared linear layer. Additionally, we extend the offsets to learnable 3D offsets $\boldsymbol{\delta}_i$ pointing to a set of 3^3 voxels. These 3^3 voxels, to which the offsets are pointing, should contain all the useful contextual information encoded by neighboring voxels. Similar to DGMN [15], we use a deformable unfold layer [22], $\Delta(\cdot,\boldsymbol{\delta}_i)$, to adapt the keys and values — $\mathbf{k}_i$, $\mathbf{v}_i$. The output of the graph reasoning is simply a 3D self-attention block over deformed keys and values:

$$\hat{\mathbf{p}}_i = \mathrm{softmax}(\mathbf{q}_i \cdot \Delta(\mathbf{k}_i,\boldsymbol{\delta}_i) + \mathrm{pos}_i) \cdot \Delta(\mathbf{v}_i,\boldsymbol{\delta}_i), \tag{5.1}$$

where $\boldsymbol{\delta}_i$=Linear($f(\mathbf{p}_i)$), $\boldsymbol{\delta}_i \in \mathbb{R}^{3\times 3^3}$, $\mathrm{pos}_i \in \mathbb{R}^{3^3}$ are the positional embeddings for the query $\mathbf{q}_i$ of input features [23]. Finally, we devoxelize $\hat{\mathbf{p}}_i$ back to point representations using the coordinate-based interpolation [20].

5.4 Experiments and results

5.4.1 Datasets

We evaluate our method on two public datasets: *MSD* [24] and *LiTS* [25]. Given that there are no Couinaud segment annotations in these two datasets, we use the annotations of Tian *et al.* [26] and Zhang *et al.* [8]. The *MSD* dataset has 192 annotated CT scans, the median in-plane resolution is 0.80 mm (min.: 0.57 mm, max.: 0.98 mm), and the median of interplanar resolutions is 5.00 mm (min.: 1.25 mm, max.: 7.50 mm). The *LiTS* dataset contains 131 annotated CT scans, with median in-plane and interplanar resolutions of 0.77 mm (min.: 0.56 mm, max.: 1.00 mm) and 1.00 mm (min.: 0.70 mm, max.: 5.00 mm), respectively.

5.4.2 Experimental setting

We reoriented all CT scans to left-posterior-inferior (LPI) and maintain the original CT image origin and spacing for all experiments. We divide both datasets into training/validation/test sets following the ratio 10/3/7. The CT values are truncated to the range of $[-100, 300]$ Hounsfield units and then normalized to $[0, 1]$. We consider each voxel, $\mathbf{v}$, of a CT scan as a point, and compute the point coordinates as: $\mathbf{p}=sd\mathbf{v}+o$ where s, d and o are the physical spacing, direction and origin parameters recorded in the CT. We also normalize the physical coordinates of the points to $[0, 1]$. We set the r in the ball-query sampling scheme (Fig. 5.3) to $\frac{1}{2\cdot 64}$ and $\frac{1}{2\cdot 32}$, and the first-scale grid size is 64^3, and 32^3 for the *MSD* and *LiTS* datasets, respectively. We randomly sample 10% points (≈50K) for each training iteration. We evaluate all methods only on the liver region, by masking out other areas. We train our model with 400 epochs and report the final number. We use a stochastic gradient descent (SGD) optimizer with a momentum of 0.98 and a learning rate of 0.01. For all the experiments, we use an NVIDIA A100 (40GB) GPU. We consider point-based baselines: PointNet [27], PointNet++ [28], AGCNN [19], and Zhang *et al.*'s [8] method, using their default settings for both training and inference.

5.4.3 Evaluation metrics

We evaluate all methods only on the liver region, by masking out other areas. We report *Dice* coefficient and average surface distance (*ASD*) in our evaluation. We use *Torchmetrics* [29] and *MONAI* [30] to calculate the two metrics. Surprisingly, these results are slightly different than reported in Zhang et al. [8]. We also report inference times and GFLOPs [31].

5.4.4 Quantitative comparison

Tab. 5.1 and Tab. 5.2 show the quantitative comparison in *Dice* coefficient and *ASD* scores. In Tab. 5.1, our method achieves the highest *Dice* score for each Couinaud segment. Both our method and *PointNet* have lower *ASD* scores compared to the other methods. *PointNet* has a low *ASD* score because, during inference, it uses a voting scheme which can reduce false positives. However, both *PointNet++* and *PointNet* have large segmentation errors for segment (I). This may be due to the segment having the lowest volume fraction in the liver.

In Tab. 5.2, our method has the highest *Dice* coefficient and the lowest *ASD* averaged over all segments. Segment (IV) is the left medial [32] section (light blue in Fig. 5.1). Segment (IV) is difficult to distinguish because of the large neighborhood area compared to the other segments. Segments (V) and (VIII) belong to the right anterior [32] section, which is adjacent to segment (IV). Thus, inaccurate segmentation of (IV) also negatively affects the accuracy of (V) and (VIII). Moreover, segment (I) is also difficult to segment due to its small volume (in pink in Fig. 5.1). This is exacerbated by the high axial resolution (*i.e.* ≤1.0 mm) of the *LiTS* dataset, which increases the voxel-wise class imbalance.

PointNet++, AGCNN, and our method use multi-scale point sampling. This results in longer inference times compared to PointNet and Zhang *et al.*, which do not employ multi-scale point sampling. The GFLOPs of AGCNN and our method depend on the radius used in the ball-query sampling.

5.4.5 Qualitative comparison

We visualize our Couinaud segmentation and the baseline segmentation on the *MSD* and *LiTS* datasets, as shown in Fig. 5.4 and Fig. 5.5. In Fig. 5.4, we show three different cases with varying axial spacing from 1.5 mm to 7.5 mm, in the *MSD* dataset. For the cases with lower axial resolution (5 mm and 7.5 mm), we show the plane in axial view, and use the red dotted bounding-box to highlight the boundaries between segments. We also show one case with relatively high axial resolution (1.5 mm) in a coronal view, in the last row of Fig. 5.4. All boxes are in the same location, in the

Table 5.1: **Quantitative evaluation on *MSD*:** We report results per segments (I – VIII), as well as the average. *PointNet* is competitive in *ASD*, however, its *Dice* coefficient is low over all segments. Our method achieves the highest average in *Dice* coefficient, and comparable average in *ASD* with *PointNet*, demonstrating the effectiveness of the added implicit knowledge. (We denote with * the use of extra vessel-priors.)

	MSD (Dice %) ↑					*MSD* (ASD mm) ↓				
	PointNet [27]	PointNet++ [28]	AGCNN [19]	Zhang *et al.* [8]	Ours	PointNet [27]	PointNet++ [28]	AGCNN [19]	Zhang *et al.* [8]	Ours
(I)	62.26	72.26	63.96	80.45	**83.31**	4.04	5.01	2.76	2.96	**1.97**
(II)	81.53	80.27	77.79	82.39	**86.71**	1.67	**1.02**	6.19	4.75	1.89
(III)	69.80	72.75	64.55	75.06	**79.35**	3.31	**1.96**	5.78	3.30	2.51
(IV)	61.74	68.04	63.26	69.83	**73.26**	5.76	5.90	6.52	5.46	**4.12**
(V)	70.55	71.78	68.78	71.49	**75.89**	**3.56**	5.61	11.75	5.91	6.36
(VI)	75.88	75.29	69.91	72.51	**79.51**	**2.68**	4.57	8.20	5.23	5.15
(VII)	82.38	82.24	80.35	80.56	**85.11**	**2.47**	3.02	5.62	5.57	3.27
(VIII)	75.86	76.03	73.72	75.74	**80.16**	**4.10**	4.19	6.79	4.82	4.84
Avg	72.50	74.83	70.29	76.00	**80.41**	**3.45**	3.91	6.70	4.75	3.76
	Time (s) per case					*GFLOPs*				
	2.81	21.55	6.13	1.73	10.03	302.38	64.65	638.54	608.85	771.15

Table 5.2: **Quantitative evaluation on *LiTS*:** Results per Couinaud segment (I – VIII) and the overall average. Zhang *et al.* [8] obtains higher scores on some segments, yet their method uses vessel priors. Our method, without prior vessel knowledge, achieves the highest average in *Dice* coefficient and the lowest average in *ASD*.

	LiTS (Dice %) ↑					*LiTS* (ASD mm) ↓				
	PointNet [27]	PointNet++ [28]	AGCNN [19]	Zhang *et al.* [8]	Ours	PointNet [27]	PointNet++ [28]	AGCNN [19]	Zhang *et al.* [8]	Ours
(I)	49.80	37.85	69.20	**73.64**	68.84	8.39	8.58	**4.36**	6.54	6.97
(II)	70.69	72.78	82.62	82.82	**86.17**	9.39	5.31	4.58	4.81	**3.18**
(III)	58.26	65.68	76.09	72.46	**80.82**	15.18	7.42	6.04	5.09	**3.18**
(IV)	53.87	70.19	73.88	**75.40**	75.24	8.46	9.69	10.00	**7.88**	8.38
(V)	80.46	80.51	78.97	81.92	**83.03**	**5.04**	5.85	7.25	6.49	6.20
(VI)	77.69	79.23	73.45	79.28	**79.29**	4.55	6.02	5.76	6.67	**3.78**
(VII)	79.92	81.71	80.42	**83.40**	82.79	**4.38**	4.93	6.75	4.63	4.60
(VIII)	77.20	79.29	77.53	80.02	**80.26**	6.45	6.67	7.71	**5.84**	7.71
Avg	68.49	70.90	76.52	78.62	**79.56**	7.73	6.81	6.56	6.00	**5.50**
	Time (s) per case					*GFLOPs*				
	10.62	106.97	19.81	6.52	13.83	302.38	64.65	256.10	608.85	141.56

Table 5.3: **Model ablation study in *Dice* coefficient and average surface distance (*ASD*) on the *MSD* [24] and *LiTS* [25] datasets**. (a) AGCNN [19] baseline; (b) without graph reasoning, $\mathscr{G}r(f(\mathbf{p}))$, in our model; (c) without grid feature embeddings, $f(\mathbf{p})$, in our proposed model; (d) our proposed model. All our model components contribute to the final model' s scores.

	MSD		*LiTS*	
	Dice ↑ (%)	ASD ↓ (mm)	Dice ↑ (%)	ASD ↓ (mm)
(a)	70.29 (± 16.69)	6.70 (± 5.80)	76.52 (± 8.90)	6.56 (± 2.65)
(b)	78.02 (± 11.06)	4.00 (± 3.39)	77.46 (± 10.87)	6.58 (± 4.03)
(c)	68.98 (± 14.58)	7.73 (± 5.19)	72.67 (± 10.30)	9.81 (± 5.44)
(d)	**80.41** (± 10.74)	**3.76** (± 4.04)	**79.56** (± 6.95)	**5.50** (± 2.42)

corresponding CT image. We can see that the boundaries of our method are most similar to the ground truth boundaries (last column). On the last row, both our method and Zhang *et al.*'s [8] boundaries follow the box centerline. However, Zhang *et al.*'s [8] method does not correctly recognize segment (III) in this case, as seen in the right-upper corner. In Fig. 5.5, we show the boundary comparison for three cases in three views (axial, coronal and sagittal). Similar to the results in Fig. 5.4, the boundaries of our segments are located on the box centerline. The baselines boundaries are slightly offset, in this case, as seen in the first row in Fig. 5.5. In the second and third rows, all baselines except ours show varying degrees of misclassification.

5.4.6 Model ablation study

To verify that all the components of our model contribute to the segmentation, we perform model ablations in Tab. 5.3. We consider four cases: (a) the baseline AGCNN [19]; (b) our method with grid feature embeddings, $f(\mathbf{p})$, but without graph reasoning, $\mathscr{G}r(f(\mathbf{p}))$; (c) our model with graph reasoning, $\mathscr{G}r(f(\mathbf{p}))$, but without grid feature embeddings, $f(\mathbf{p})$; (d) our complete model. Grid feature embeddings $f(\mathbf{p})$ are an important bridge between features in 3D point-space, $\mathbf{p}$, and the graph reasoning module $\mathscr{G}r(f(\mathbf{p}))$. As seen for the setting (c) in Tab. 5.3, the graph reasoning module fails to work directly, on the voxelized point-features. In addition, the graph reasoning module in itself contributes to the model predictions, as seen in the setting (b) of Tab. 5.3. Removing $\mathscr{G}r(f(\mathbf{p}))$ decreases the dice scores by $\approx 2\%$ on both the *MSD* and *LiTS* datasets. On both the *LiTS* data and the *MSD* dataset, all model components prove useful.

5.5 Discussion

Although the proposed method does not need an explicit vessel prior as [8, 11], it still depends on the anatomical landmarks in the liver to learn graph structures as

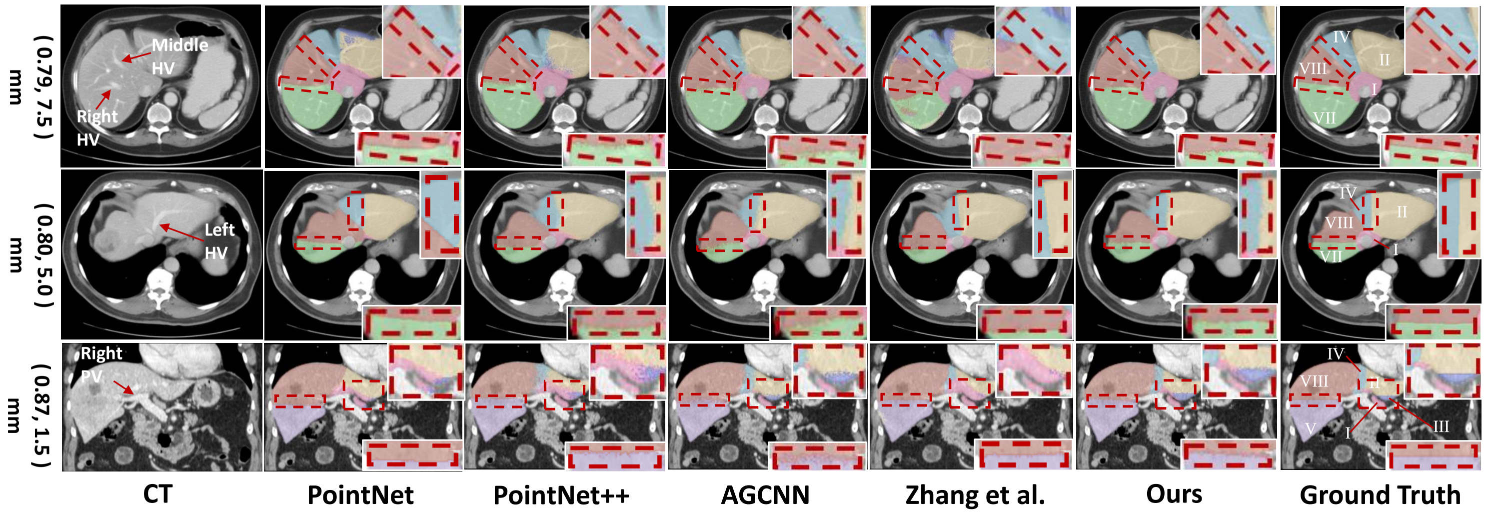

Figure 5.4: **Qualitative evaluation on *MSD*.** The anatomical landmarks [32] are marked with red arrows in the CT image, as a reference. The red dotted bounding-box highlights the boundary between segments. Our method predicts segment boundaries that are closest to the ground truth. We show to the left the in-plane and interplane resolutions. (HV: heaptic vein, PV: portal vein)

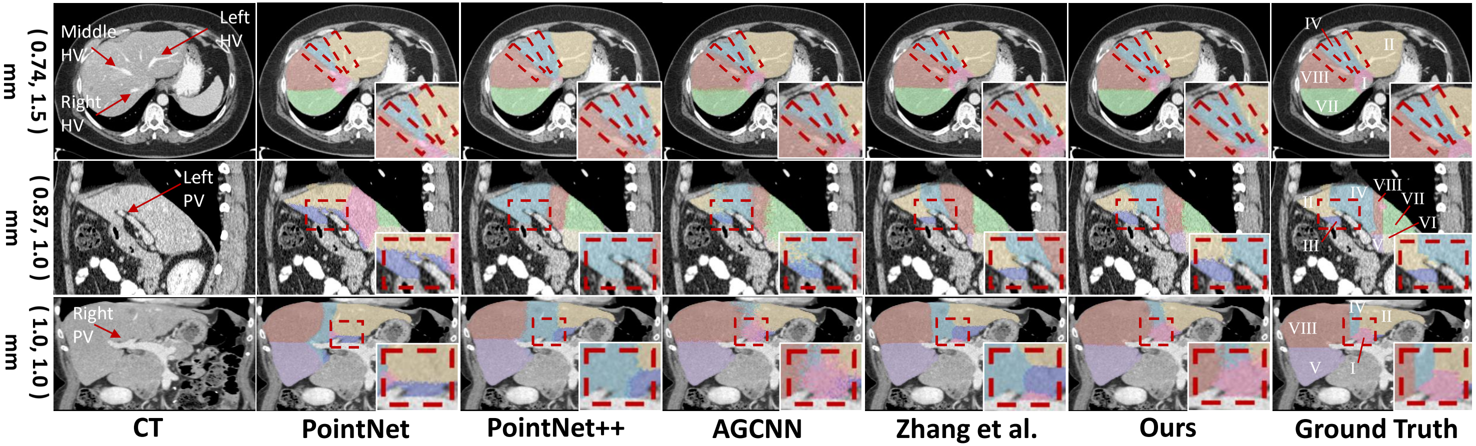

Figure 5.5: **Qualitative evaluation on *LiTS*.** Three cases in axial, sagittal, and coronal views. The red arrows mark the anatomical landmarks [32], and the bounding-boxes highlight segment boundaries. Our method displays the most accurate boundaries.

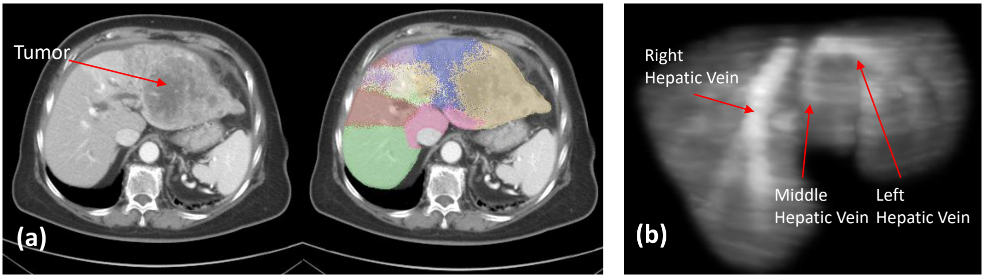

Figure 5.6: **Failure case analysis.** (a) The anatomical landmarks disappear in the tumor region marked by the red arrow. (b) Output of $\mathscr{G}r(f(\mathbf{p}))$ of a successful case. The model learns the graph affinities based on the anatomical landmarks in the CT images.

shown in Fig. 5.6 (b). Therefore, the proposed method cannot accurately discriminate segments in the liver when a large tumor appears, as shown in Fig. 5.6 (a). This is because the partial anatomical landmarks in the CT image are absent, such as the left hepatic vein and the left portal vein. If no anatomical landmarks are visible, the model divides the liver based on noise in the CT images, resulting in segmentations without well-defined borders. In addition, a relatively larger liver with a large tumor, as in Fig. 5.6, also causes inaccurate segmentation. This is because there is an obvious difference between the 3D coordinates of a normal-sized liver and those of a larger liver with a tumor. The proposed point-based method depends on the 3D coordinates when performing the inference, coordinates outliers may fail the discrimination of the Couinaud segment boundary.

5.6 Conclusion

We propose a 3D point-based method for Couinaud segmentation for CT images. Unlike prior work, we do not rely on anatomic prior knowledge of the vessel structure in the liver area. Our model learns implicitly contextual anatomic information, by learning affinities between voxels in the CT volume in a dynamic graph reasoning module. This implicit contextual information makes our model competitive compared to state-of-the-art, where we exceed PointNet, PointNet++, AGCNN and Zhang *et al.*'s [8] method on the *MSD* and *LiTS* public benchmarks. This approach voids the time-consuming definition of the vessel structure in the liver while still showing the better performance.

References

[1] J. M. Llovet, R. K. Kelley, A. Villanueva, et al. “Hepatocellular carcinoma (primer)”. In: *Nature Reviews: Disease Primers* 7.1 (2021), page 6.

[2] O. I. Alirr and A. A. A. Rahni. “Survey on liver tumour resection planning system: steps, techniques, and parameters”. In: *Journal of Digital Imaging* 33.2 (2020), pages 304–323.

[3] M. Luo, H. Jiang, and T. Shi. “Multi-stage puncture path planning algorithm of ablation needles for percutaneous radiofrequency ablation of liver tumors”. In: *Computers in Biology and Medicine* 145 (2022), page 105506.

[4] Z. Arya, G. Ridgway, A. Jandor, and P. Aljabar. “Deep learning-based landmark localisation in the liver for Couinaud segmentation”. In: *Medical Image Understanding and Analysis: 25th Annual Conference, MIUA 2021, Oxford, United Kingdom, July 12–14, 2021, Proceedings 25*. Springer. 2021, pages 227–237.

[5] X. Jia, C. Qian, Z. Yang, et al. “Boundary-aware dual attention guided liver segment segmentation model”. In: *KSII Transactions on Internet and Information Systems (TIIS)* 16.1 (2022), pages 16–37.

[6] J. Tian, L. Liu, Z. Shi, and F. Xu. “Automatic couinaud segmentation from CT volumes on liver using GLC-UNet”. In: *International workshop on machine learning in medical imaging*. Springer. 2019, pages 274–282.

[7] M. Wang, R. Jin, J. Lu, et al. “Automatic CT liver Couinaud segmentation based on key bifurcation detection with attentive residual hourglass-based cascaded network”. In: *Computers in Biology and Medicine* 144 (2022), page 105363.

[8] X. Zhang, S. Ali, T. Liu, et al. “Robust and smooth Couinaud segmentation via anatomical structure-guided point-voxel network”. In: *Computers in Biology and Medicine* 182 (2024), page 109202.

[9] D. Boltcheva, N. Passat, V. Agnus, et al. “Automatic anatomical segmentation of the liver by separation planes”. In: *Medical Imaging 2006: Visualization, Image-Guided Procedures, and Display*. Volume 6141. SPIE. 2006, pages 383–394.

[10] S. Pla-Alemany, J. A. Romero, J. M. Santabárbara, et al. “Automatic multi-atlas liver segmentation and Couinaud classification from CT volumes”. In: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2021, pages 2826–2829.

[11] X. Zhang, Y. Liu, S. Ali, et al. “Anatomical-aware point-voxel network for couinaud segmentation in liver ct”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2023, pages 465–474.

[12] Y. Chen, M. Rohrbach, Z. Yan, et al. “Graph-based global reasoning networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pages 433–442.

[13] F. Manessi, A. Rozza, and M. Manzo. “Dynamic graph convolutional networks”. In: *Pattern Recognition* 97 (2020), page 107000.

[14] X. Wang, R. Girshick, A. Gupta, and K. He. “Non-local neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pages 7794–7803.

[15] L. Zhang, M. Chen, A. Arnab, et al. “Dynamic Graph Message Passing Networks for Visual Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[16] Y. Ma, Y. Guo, H. Liu, et al. “Global context reasoning for semantic segmentation of 3D point clouds”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pages 2931–2940.

[17] Y. Wang, Y. Sun, Z. Liu, et al. “Dynamic graph cnn for learning on point clouds”. In: *ACM Transactions on Graphics (tog)* 38.5 (2019), pages 1–12.

[18] W. Zhou, Q. Wang, W. Jin, et al. “Graph Transformer for 3D point clouds classification and semantic segmentation”. In: *Computers & Graphics* 124 (2024), page 104050.

[19] M. Wei, Z. Wei, H. Zhou, et al. “AGConv: Adaptive graph convolution on 3D point clouds”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.8 (2023), pages 9374–9392.

[20] Z. Liu, H. Tang, Y. Lin, and S. Han. “Point-voxel cnn for efficient 3d deep learning”. In: *Advances in neural information processing systems* 32 (2019).

[21] Z. Liu, Y. Lin, Y. Cao, et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pages 10012–10022.

[22] X. Zhu, H. Hu, S. Lin, and J. Dai. “Deformable convnets v2: More deformable, better results”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pages 9308–9316.

[23] P. Shaw, J. Uszkoreit, and A. Vaswani. “Self-Attention with Relative Position Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Edited by M. Walker, H. Ji, and A. Stent. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pages 464–468.

[24] M. Antonelli, A. Reinke, S. Bakas, et al. “The medical segmentation decathlon”. In: *Nature communications* 13.1 (2022), page 4128.

[25] P. Bilic, P. Christ, H. B. Li, et al. “The liver tumor segmentation benchmark (lits)”. In: *Medical Image Analysis* 84 (2023), page 102680.

[26] J. Tian, L. Liu, Z. Shi, and F. Xu. “Automatic couinaud segmentation from CT volumes on liver using GLC-UNet”. In: *International workshop on machine learning in medical imaging*. Springer. 2019, pages 274–282.

[27] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. “Pointnet: Deep learning on point sets for 3d classification and segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pages 652–660.

[28] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. “Pointnet++: Deep hierarchical feature learning on point sets in a metric space”. In: *Advances in neural information processing systems* 30 (2017).

[29] N. S. Detlefsen, J. Borovec, J. Schock, et al. “Torchmetrics-measuring reproducibility in pytorch”. In: *Journal of Open Source Software* 7.70 (2022), page 4101.

[30] M. J. Cardoso, W. Li, R. Brown, et al. “Monai: An open-source framework for deep learning in healthcare”. In: *arXiv preprint arXiv:2211.02701* (2022).

[31] xiaoju ye. *calflops: a FLOPs and Params calculate tool for neural networks in pytorch framework*. 2023. URL: `https://github.com/MrYxJ/calculate-flops.pytorch`.

[32] S. Rutkauskas, V. Gedrimas, J. Pundzius, et al. “Clinical and anatomical basis for the classification of the structural parts of liver”. In: *Medicina (Kaunas)* 42.2 (2006), pages 98–106.