Deep learning for vascular segmentation and tissue characterization in CT images

Zhang, X.

3

Continuous and complete liver vessel segmentation with graph-attention guided diffusion

This chapter was adapted from:

Zhang, X., Broersen, A., van Erp, G., Pintea, S.L. and Dijkstra, J., **Continuous and complete liver vessel segmentation with graph-attention guided diffusion.** (2025) Knowledge-Based Systems, 331, Article 114686.

Abstract

Improving connectivity and completeness are the most challenging aspects of liver vessel segmentation, especially for small vessels. These challenges require both learning the continuous vessel geometry and focusing on small vessel detection. However, current methods do not explicitly address these two aspects and cannot generalize well when constrained by inconsistent annotations. Here, we take advantage of the generalization of the diffusion model and explicitly integrate connectivity and completeness in our diffusion-based segmentation model. Specifically, we use a graph-attention module that adds knowledge about vessel geometry. Additionally, we perform the graph-attention at multiple-scales, thus focusing on small liver vessels. Our method outperforms eight state-of-the-art medical segmentation methods on two public datasets: *3D-ircadb-01* and *LiVS*.

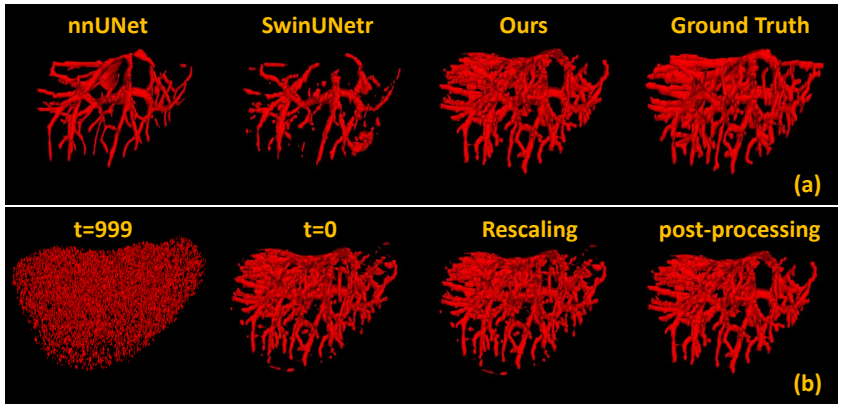


Figure 3.1: (a) Vessel trees of *nnUNet* [1], *Swin UNETR* [2], our proposed model and the ground truth. (b) Vessel trees predicted by our method across different steps (t=0 is the final diffusion iteration). Rescaling is the process of recovering the resolution of resized CT images to that of the original CT images.

3.1 Introduction

Liver cancer is the fourth leading cause of death according to statistics on cancer-related mortality [3]. Furthermore, the liver is a frequent site for metastasis of various primary tumors, such as gastrointestinal tumors, breast cancer, lung cancer, and melanoma [4]. Both primary and secondary liver cancer have multiple treatment options, including surgery, and various interventional oncology liver treatments. The preoperative planning of these treatments can be improved with accurate segmentation of the liver vessels [5, 6]. In preoperative planning of liver tumor resection [6], visualizing the spatial location between liver vessels and tumors in a 3D view is essential to reduce surgical risk. Liver vessel segmentation is used primarily to ensure that the main vessels are not located near the planned resection region, thus reducing bleeding. Liver vessels also indicate the boundaries for the Couinaud classification [6]. Furthermore, it can assist in targeting the correct tumor nutrient supply vessel to decrease the recurrence rate in embolic therapies [7]. Hence, accurate liver vessel segmentation is essential in liver tumor surgery. However, acquiring automatic liver vessel segmentation is challenging due to the complex anatomy.

Automatic liver vessel segmentation is performed on computed tomography (CT) images. Traditionally, methods relied on image filtering [8, 9], active contour models [10, 11], or tracking methods [12, 13]. Currently, the leading methods on liver vessel segmentation are based on deep artificial networks [14, 15, 1, 2]. *nnUNet* [1] can extract features automatically from the CT images. However, *nnUNet* cannot ensure vessel continuity, as shown in Fig. 3.1(a). Attention-based methods

[16, 17] as used in *Swin UNETR* [2] improve vessel continuity. Yet, *Swin UNETR* struggles with horizontally distributed vessel, as seen in Fig. 3.1(a). In addition, the performance of the above Convolutional Neural Network (CNN) based methods have the generalization problem, especially when the label annotation styles across data are different [18]. Thus, ensuring vessel continuity in all directions, localizing small vessels, and improving model generalization for the inconsistent annotation style remains challenging.

Here, we address these challenges by proposing a model that makes continuous, and complete predictions, and generalizes well to different annotation styles. The inherent anatomical structure of the vascular tree inspires our network design. We are motivated by the assumption that introducing an explicit vascular graph can help the model better capture irregular and long-range vessel connectivity, thus adding continuity. Additionally, we use multiple scales in our graph structure, thus enabling the detection of small vessels, adding completeness to the model. Moreover, prior work tends to overlook the underlying data distribution, resulting in a strong dependence on annotation quality. To achieve accurate segmentation despite imperfect annotations, we rely on a diffusion model to learn the underlying data distribution, and mitigate the dependence on annotation quality.

Concretely, our model starts from a 2*D* diffusion model [19, 20]. We opt for a 2D rather than a 3D diffusion model to reduce computational requirements. To ensure vessel continuity, we add graph-attention layers [21] into the diffusion model. Because the graph is sparse, we compensate for this by integrating neighboring features on the graph in a local ensemble module [22]. The local ensemble module ensures a smooth transition between different nodes [22]. Secondly, to segment small vessels, we extract features at multiple scales in the nodes of the graph. The effectiveness of these components is shown in Fig. 3.1(b).

Our contributions are: (i) explicitly incorporating vessel continuity by adding graph-attention conditioning to a diffusion model for liver vessel segmentation; (ii) explicitly focusing on small vessels by relying on multi-scale graph-features when conditioning the diffusion model; (iii) continuous and complete vessel segmentations on two public datasets *3D-ircadb-01* [23] and *LiVS* [24], compared to existing work.

3.2 Related Work

3.2.1 Liver vessel segmentation

Liver vessel segmentation currently relies on *CNN* and attention methods. CNN methods either rely on *FCN* (fully convolutional networks) [15, 25], or follow the *UNet* architecture [14, 26]. Attention methods can be grouped into self-attention [16, 27] and graph-attention [28, 17] methods. *UNet* [29] and its variants are effective for medical segmentation. However, they struggle with imbalanced data [30],

such as between the vascular and the liver region. Moreover, their performance is limited by the receptive-field size of the convolutions [31]. While the data imbalance can be mitigated by improving the *Dice* loss[14], learning long-range dependencies remains a challenge [2]. To address this, *Swin-transformer* [32] uses a transformer architecture, and thus its receptive-field size covers the full input resolution. Similarly, graph-attention networks [33] capture long-range dependencies by computing node attention-coefficients. Augmenting the *UNet* by *Swin-transformer* [16], or using the graph-attention to assist the *UNet* training[17] is also effective in practice. In addition to methods specifically developed for liver vessel segmentation, several works on retinal vessel segmentation [34], skin lesion segmentation [35], and general medical image segmentation [36, 37] also incorporate self-attention mechanisms or U-Net variants to improve segmentation accuracy. Here, we also combine the capabilities of *CNNs* and attention mechanisms. Moreover, we jointly enforce continuity of the vessel-tree segmentations and focus on small vessels.

3.2.2 Diffusion models for medical image segmentation

Diffusion [19] methods showcase promising results for medical image segmentation. These diffusion methods are either non-dynamic conditioning [38] or dynamic conditioning [39, 40]. The non-dynamic models concatenate medical images to the input, and do not adapt this conditioning information over time. Their performance [38, 41] is comparable to (or lower than) *nnUNet* [1], which is the standard medical segmentation baseline. On the other hand, dynamic conditioning methods [40] use an extra encoder to generate time-dependent conditioning information. Similarly to the non-dynamic conditioning models, their accuracy is limited. More recently, *HiDiff* [42] and *MedSegDiff* [39] using a hybrid constrained method, obtains state-of-the-art results. *MedSegDiff* predicts a weighted combination of a diffusion segmentation and an auxiliary segmentation from the conditioning branch. *HiDiff* [42] also relies on a prior segmentation as one of the conditioning inputs to guide the diffusion model. This combination weakens the contribution of the diffusion model. Here, we build on a 2D dynamic conditioning diffusion model constrained by a hybrid loss, yet we only predict the diffusion segmentation.

3.2.3 Graph-based methods for medical image segmentation

Graphs are one of the most intuitive way to represent complex anatomical structures. Graph-based methods can segment medical tree structures [43, 44], but also non-structural medical data [45, 46]. Although, the graphs add long-range dependencies in CNNs, they tend to miss small branches [47]. This is due to the sparsity of the nodes, causing loss of information. Here, we also rely on graphs to add connectivity for vessel segmentation. To avoid missing small vessels, we use the local ensemble module of LIIF [22] which smoothes the feature between nodes. Additionally, we use

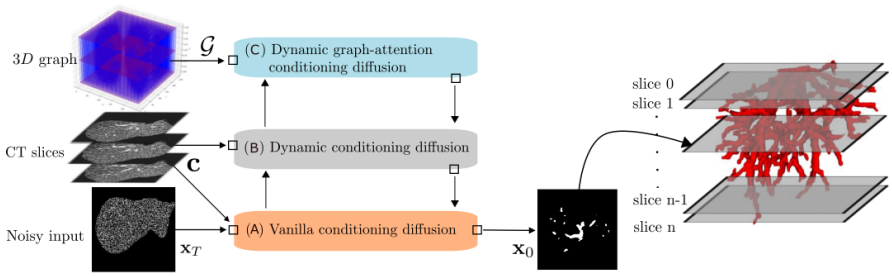


Figure 3.2: **Overview of our model:** (A) A vanilla diffusion model starting from noisy images $\mathbf{x}_T$, and predicting segmentation masks $\mathbf{x}_0$ (in orange); (B) A dynamic conditioning model, conditioned on three CT slices $\mathbf{c}$ (in gray); and (C) A multiscale graph-attention conditioning model, starting from a graph structure $\mathcal{G}$ (in blue).

multiscale graph features to focus on small vessels.

3.3 Methods

3.4 Diffusion conditioning models

Our model contains three components, as show in Fig. 3.2: (A) The vanilla conditioning diffusion model over three CT slices, for conditioning; (B) Dynamic conditioning diffusion, starting from the same CT slices but using a separate encoder; and (C) Conditioning diffusion model with multiscale graph-attention guidance.

3.4.1 Vanilla conditioning diffusion model

Diffusion model. Conditioning diffusion models extend the Denoising Diffusion Probabilistic Models (DDPM) [19]. DDPM is composed of a forward process and a reverse process.

The forward process gradually adds Gaussian noise to the inputs $\mathbf{x}_0$ over a number of T timesteps. In our case $\mathbf{x}_0$ is the ground truth vessel segmentation mask. The variance of the Gaussian noise is modeled by β_t, which is typically a linear function of t. Thus, the distribution of the noisy vessel mask $\mathbf{x}_t$ given the ground truth $\mathbf{x}_0$, is:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)I), \tag{3.1}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$. In the forward process, we obtain the noisy vessel mask $\mathbf{x}_t$ from $\mathbf{x}_0$ as a linear combination with the noise $\epsilon_t \sim \mathcal{N}(0, I)$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t. \tag{3.2}$$

In the reverse process, we train a model p_θ with parameters θ, to iteratively denoise an input noisy image $\mathbf{x}_T$. This aims to recover the clean segmentation mask $\mathbf{x}_0$. The

model p_θ follows a Gaussian distribution:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)). \tag{3.3}$$

The mean $\mu_\theta(\mathbf{x}_t, t)$ and variance $\Sigma_\theta(\mathbf{x}_t, t)$ of the reverse process are functions of a noise model $\epsilon_\theta(\mathbf{x}_t, t)$. The noise ϵ_θ is typically modelled by a *UNet* [29, 14]. This *UNet* noise model ϵ_θ is trained by minimizing the difference between the estimated noise ϵ_θ and the true noise ϵ_t at a number of sampled timesteps $t{\sim}[1, T]$:

$$L_{\text{den}}(\mathbf{x}_0, \theta) = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\epsilon_t} \|\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t)\|^2. \tag{3.4}$$

Conditioning diffusion models. Following [38], we add a conditioning to the DDPM model that is independent of the timestep t. We use three consecutive CT slices, $\mathbf{c}$, as conditioning for our DDPM model. We evaluate the vanilla conditioning model in the experiments.

3.4.2 Dynamic Conditioning model

Vanilla conditioning cannot adapt the condition across the noise levels (time steps). To address this, at every diffusion timestep t, we embed the CT slices, $\mathbf{c}$, by using the *GenericUNet* encoder from *nnUNet* [1], giving rise to $\mathbf{f}_\mathbf{c}^t$. To obtain time-dependent conditioning, $\mathbf{f}_\mathbf{c}^t$, at each timestep t into the bottleneck of the *vanilla conditioning diffusion model*, as shown in Fig. 3.3(B) (solid downward arrow). Additionally, we use group convolutions in the conditioning, thus keeping the features per slice separate. To let the CT embedding $\mathbf{f}_\mathbf{c}^t$ adapt over time, we merge the noisy features of the *vanilla conditioning model* into the *dynamic conditioning model*, at the appropriate depth (in Fig. 3.3(B) with dashed arrows).

3.4.3 Multiscale graph-attention conditioning model

The *vanilla conditioning* and the *dynamic conditioning* both use the denoising loss in Eq. (3.4). To make use of the geometric structure of vessels, we first map the 3D vessel tree into a graph $\mathcal{G}$. Subsequently, we use graph-attention [21] to add this geometric structure as a condition into the diffusion model.

Vessel graph construction. We construct a 3D vessel graph $\mathcal{G}{=}(\mathbf{V}, \mathbf{E})$, where the nodes $\mathbf{V}$ are locations along the vessel, and the edges $\mathbf{E}$ indicate vascular connectivity. We start from the full volume $[D{\times}H{\times}W]$ of a ground truth vessel tree, and we split it into non-overlapping sub-volumes $[d{\times}h{\times}w]$. Each node $\mathbf{v}{\in}\mathbf{V}$ corresponds to a sub-volume and is the average of the voxel coordinates along the vessel region. If there is no vascular annotation in a sub-volume, we use the central voxel as the node. The graph edges, $\mathbf{e}{\in}\mathbf{E}$, are the geodesic distances between nodes, as in the Vessel Graph Network of Shin et al. [44]. Nodes with a small distance, but belonging to different vessel

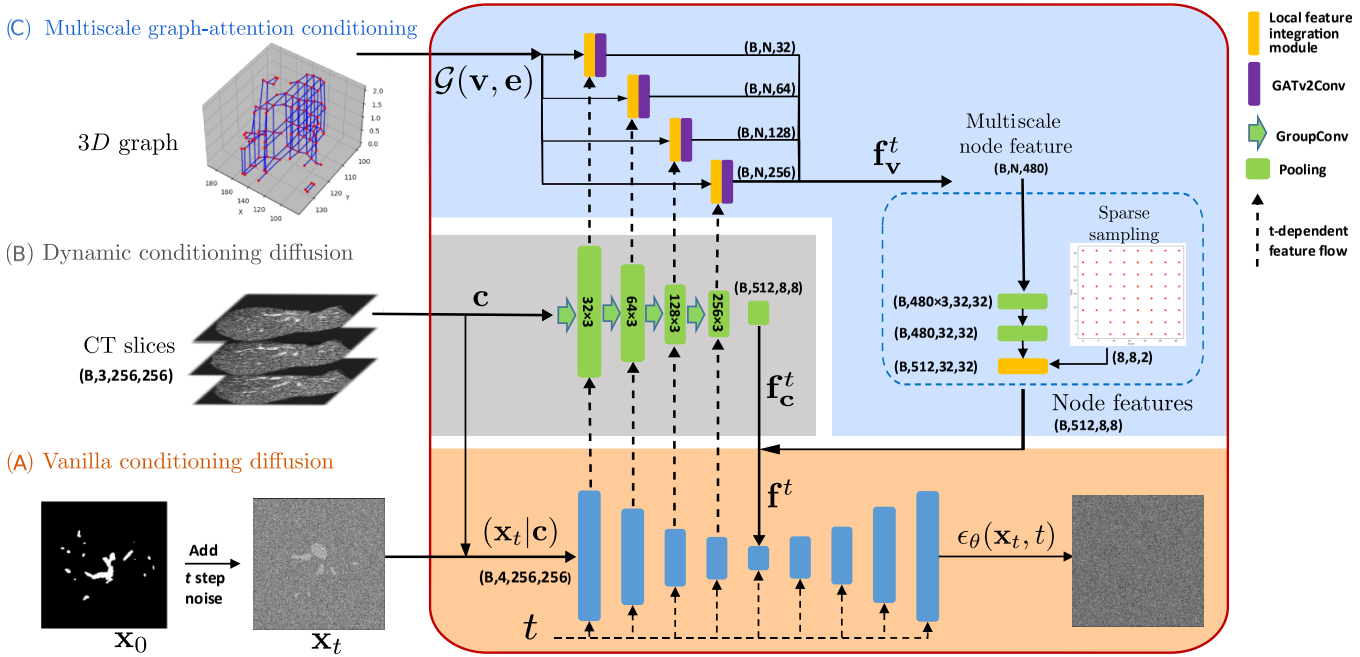


Figure 3.3: **Our network architecture:** (A) vanilla conditioning diffusion (orange); (B) dynamic conditioning diffusion (gray); (C) multiscale graph-attention conditioning (blue).
These components interact through the vertical dashed/solid arrows. The dashed upwards arrows adapt the conditioning over time.
The solid downwards arrows add the conditioning features: CT slices $\mathbf{c}$, CT slice embeddings $\mathbf{f}_\mathbf{c}^t$, and graph embeddings $\mathbf{f}_\mathbf{v}^t$.

branches, should not be connected. For this, we use the binary vessel label as a speed function to calculate the travel time from one node to another, as in Li et al. [17].

Graph training and inference. During training, we construct the graph $\mathscr{G}$ using the ground truth vessel masks, as in Fig. 3.3(C). We only display the foreground nodes, corresponding to the location of the vessels, and leave the background nodes transparent. During inference, we do not have access to the ground truth vessel masks. Therefore, we input to the *multiscale graph-attention conditioning model* a fully-connected graph, as in Fig. 3.2(C). This is a viable choice, because during training, the graph helps adapt the weights of the component (C). Specifically, via the graph-attention, it extracts informative features from component (B). At inference, these attention-weights are trained, and can be applied on the new input features coming from component (B). In the ablation studies we show the effectiveness of using a fully-connected graph at inference.

Multiscale graph-attention. We use the CT-slice embeddings, $\mathbf{f}_{\mathbf{c}}^{t}$, from the *dynamic conditioning model* to extract node features at each timestep t (dashed arrows in Fig. 3.3(C)). We process the node features via a graph-attention layer GATv2 [21] and a local feature integration module (LIIF) [22], to obtain node attention-coefficients. These node attention-coefficients are concatenated over the different scales (network depths), coming from the *dynamic conditioning model*, giving rise to multiscale node attention-coefficients: $\mathbf{f}_{\mathbf{v}}^{t}$ at each node $\mathbf{v}$ and timestep t.

The nodes in the vessel graph $\mathscr{G}$ are sparse (only 32×32 nodes for one CT slice). To compensate for this sparsity, we use the local features integration module LIIF [22], which is popular for image super-resolution. We extend LIIF from 2D to 3D, and apply it on our CT-embeddings $\mathbf{f}_{\mathbf{c}}^{t}$. Specifically, for a CT-slice embedding $\mathbf{f}_{\mathbf{c}}^{(t,i)}$ at timepste t and location i corresponding to a graph node $\mathbf{v}_i$, we use the graph neighboring locations $\mathbf{v}_{ne(i)}$ to define a new embedding:

$$\hat{\mathbf{f}}_{\mathbf{c}}^{(t,i)} = \sum_{ne(i)} \frac{\mathscr{S}(\mathbf{v}_i, \mathbf{v}_{ne(i)})}{\overline{\mathscr{S}}} \, \mathrm{LFI}\left(\mathbf{f}_{\mathbf{c}}^{(t,ne(i))}, \mathbf{v}_i - \mathbf{v}_{ne(i)}\right), \tag{3.5}$$

where our neighboring locations $ne(i)$ vary across x, y, and z directions, rather than just x, y; and $\mathscr{S}(\cdot,\cdot)$ computes the area a 3D cube between the graph node $\mathbf{v}_i$ and its neighbor $\mathbf{v}_{ne(i)}$; $\overline{\mathscr{S}} = \sum_{ne(i)} \mathscr{S}(\mathbf{v}_i, \mathbf{v}_{ne(i)})$; $\mathbf{f}_{\mathbf{c}}^{(t,ne(i))} = \mathrm{GridSample}(\mathbf{f}_{\mathbf{c}}^{t}, ne(i))$ [48]; and the local feature integration module is $\mathrm{LFI}\left(\mathbf{f}_{\mathbf{c}}^{(t,ne(i))}, \mathbf{v}_i - \mathbf{v}_{ne(i)}\right) = \mathrm{Conv2d}\left(\mathrm{Cat}\left(\mathbf{f}_{\mathbf{c}}^{(t,ne(i))}, \mathbf{v}_i - \mathbf{v}_{ne(i)}\right)\right)$. Inside the function $\mathrm{LFI}(\cdot,\cdot)$ [22], we extract the neighboring features of each node based on their 3D coordinates $\mathbf{v}_{ne(i)}$, and embed the relative position differences $(\mathbf{v}_i - \mathbf{v}_{ne(i)})$, which represent the spatial relationships among neighborhood features.

Subsequently, we use new CT-slice embeddings $\hat{\mathbf{f}}_{\mathbf{c}}^{(t,i)}$ to extract node features via a

graph-attention layer GATv2 [21], for every two neighboring locations i, j:

$$\mathbf{f}_{\mathbf{v}}^{(t,i,j)} = \frac{\exp\left[\mathbf{a}^{\top}\text{LeakyReLU}\left(\mathbf{W}(\hat{\mathbf{f}}_{\mathbf{c}}^{(t,i)} + \hat{\mathbf{f}}_{\mathbf{c}}^{(t,j)})\right)\right]}{\sum_{j'}\exp\left[\mathbf{a}^{\top}\text{LeakyReLU}\left(\mathbf{W}(\hat{\mathbf{f}}_{\mathbf{c}}^{(t,i)} + \hat{\mathbf{f}}_{\mathbf{c}}^{(t,j')})\right)\right]}. \tag{3.6}$$

During training, we want to learn which CT embeddings correspond to the foreground vessels and which not. Therefore, we process the node features via a convolutional layer with sigmoid activation, to obtain $\hat{\mathbf{f}}_{\mathbf{v}}^{t}$=sigmoid(Conv($\mathbf{f}_{\mathbf{v}}^{t}$)), and optimize the model parameters θ using a binary cross-entropy loss:

$$L_{\text{graph}}(\mathcal{G},\theta) = -\sum_{\mathbf{v}\in\mathcal{G}}\sum_{t}\log(\hat{\mathbf{f}}_{\mathbf{v}}^{t}). \tag{3.7}$$

3.4.4 Overall diffusion model conditioning

We use the multiscale graph-attention embeddings $\mathbf{f}_{\mathbf{v}}^{t}$ together with the CT embeddings, $\mathbf{f}_{\mathbf{c}}^{t}$, to condition the reverse diffusion process. Therefore, Eq. (3.3) defining the reverse process, becomes:

$$p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1};\mu_{\theta}(\mathbf{x}_t|\mathbf{f}^t,t),\Sigma_{\theta}(\mathbf{x}_t|\mathbf{f}^t,t)), \tag{3.8}$$

$$\text{where } \mathbf{f}^t = \mathbf{f}_{\mathbf{c}}^{t} + \mathbf{f}_{\mathbf{v}}^{t}. \tag{3.9}$$

3.4.5 Overall loss function

Our overall loss function, used to fit the model parameters θ, is a combination of the denoising loss in Eq. (3.4) and graph loss in Eq. (3.7):

$$L_{\text{total}}(\mathbf{x}_0,\mathbf{c},\mathcal{G},\theta) = L_{\text{den}}(\mathbf{x}_0,\mathbf{c},\mathcal{G},\theta) + L_{\text{graph}}(\mathcal{G},\theta), \tag{3.10}$$

where the denoising loss L_{den} relies on the reverse diffusion process in Eq. (3.9).

3.5 Experiments and results

3.5.1 Datasets

We use two public datasets: *3D-ircadb-01* [23] and *LiVS* [24], as detailed in Tab. 3.1. *3D-ircadb-01* contains 20 cases, while *LiVS* contains 532 cases. In the *3D-ircadb-01* dataset every slice is annotated, but some small vessels are not annotated [26]. In the *LiVS* dataset only a subset of randomly chosen slices are annotated.

CT scans with thick slices (≥2.5 mm for *3D-ircadb-01* and ≥5 mm for *LiVS*) are rare, forming outliers. Thus, for *LiVS* we exclude these cases from training and test. For the relatively small *3D-ircadb-01* dataset, we only exclude the cases with thick slices from the testset. To effectively evaluate on the small *3D-ircadb-01* dataset, we asked a clinical expert to score the completeness of the annotated vessels. After thickness exclusion, cases $\{04,06,08,11,16\}$ were marked as complete. Therefore, we use these for testing. Specifically, we perform leave-one-out cross-validation on these

five cases, where at every fold we train on 19 cases and test on 1 test sample, from the list above. We average the results over all folds. For *LiVS* we report average metrics over 3-fold cross-validation.

3.5.2 Data pre- and post-processing

For the *3D-ircadb-01* [23] dataset, we first crop the liver region and resize the cropped CT slices to 256×256 px. The liver masks exclude the *vena cava*. We clip the intensity of the CT slices to $[0, 400]$ HU (Hounsfield units). The CT slices are already cropped, resized, and clipped in the *LiVS* [24] dataset. Because not all *LiVS* slices are annotated, we use ITK-SNAP[49] to interpolate the annotations. In our method, we sample the central slice of the 2.5D block only from the set of CT slices with ground truth annotations.

During inference, we rescale the diffusion predictions back to the physical resolution of the original CT image of 512×512 px. During post-processing, we remove disconnected noisy spots, with a volume less than 1% of the largest connected region, using connected region analysis [50]. When the vessel annotations are continuous in the longitudinal direction, we find post-processing [14] more effective to obtain the final vessels. For discontinuous annotations, ensemble inference with different seeds, is more effective.

3.5.3 Evaluation metrics

For all our experiments we report: Dice similarity coefficient (*DSC*), voxel-wise sensitivity (*Sen*), voxel-wise specificity (*Spe*) [51], centerline Dice (*clDice*) [52], and a custom connected region-wise connectivity (*Con*) following Gegúndez-Arias *et al.* [53]. The *Con* metric is the ratio of the total number of connected regions, in the predicted tree $\mathcal{T}$ and the total number of connected regions in the ground truth vessel

Table 3.1: Dataset overview. In *3D-ircadb-01* [23] every CT slice is annotated, but some small vessels are missing with an inconsistent annotation style. *LiVS* [24] contains more CT volumes and has stable annotation style, but only a subset of the slices are annotated.

	3D-ircadb-01	*LiVS*
Available scans	20	532
Used scans	20	303
Exclusion	completeness score CT thickness ≥2.5 mm	#annotations (< 30 slices) CT thickness ≥2.5mm
Pixel spacing	0.57 mm – 0.87mm	0.51 mm – 0.98 mm
Slice thickness	1.00 mm – 4.00mm	0.62 mm – 5.00 mm
Continuous annotation	Yes	No
High-contrast tumors	No	Yes
Annotation consistency	Low	High

tree $\mathscr{T}^*$:

$$\text{Con}(\mathscr{T},\mathscr{T}^*) = \frac{|\text{comp}(\mathscr{T})|}{|\text{comp}(\mathscr{T}^*)|} \geq 1, \tag{3.11}$$

where comp$(\cdot)$ computes the connected components. We only consider connected regions with a volume greater than 120 mm^3, as in Huang *et al.* [14]. We exclude over-connected segmentations, where Con<1.0.

3.5.4 Experimental setting

Baseline models. We compare our model with seven state-of-the-art medical segmentation methods, including three diffusion-based methods (*HiDiff*[42], *MedSegDiff*[39], *EnsemDiff*[38]), two self-attention methods *MERIT*[54] and *Swin UNETR* [2], one self-configuring method *nnUNet* [1] and one specific liver vessel segmentation method [24]. The *HiDiff*, *MERIT*, *MedSegDiff* and *EnsemDiff* use inputs of size [3×256×256], while *Swin UNETR* and *nnUNet* are 3D methods starting from the cropped CT slices as input. *Swin UNETR*, *EnsemDiff* and *MedSegDiff* require ensembles. For *Swin UNETR*, we train 5 models and ensemble their segmented results. For *EnsemDiff* and *MedSegDiff*, we ensemble 5 diffusion results using different random seeds, but with a single training. Our method does not use ensembles when training with continuous annotations on *3D-ircadb-01*, but uses 5× ensembles for discontinuous annotations on *LiVS*.

Implementation details. We perform all our experiments on 1 NVIDIA RTXA6000 GPU with 48 GB memory. We use the *AdamW* optimizer with an initial learning rate of 1×10^{-4} and a batch size of 10. We input 4 channels: namely, three CT-slices and one noisy ground truth at each time step t during training, and we use a random Gaussian noise channel during inference. Our model converges within 160k iterations. We train the *EnsemDiff* and *MedSegDiff* models following their official implementation for 60k[38] and 100k[39], respectively. We also train the *MERIT*[54] and *HiDiff*[42] using their official implementations and recommended configurations. On both the *3D-ircadb-01* and *LiVS* datasets, we use the standard DDPM sampling scheme [20] with 1000 denoising steps, during inference, for all the diffusion-based experiments. For our model, the CT block size is [3×256×256] and the vessel graph $\mathscr{G}$ consists of nodes of size $(N,3)$ and edges of size $(E,2)$. Where we set the number of graph nodes N is 32×32 per CT slice.

3.5.5 Quantitative comparison

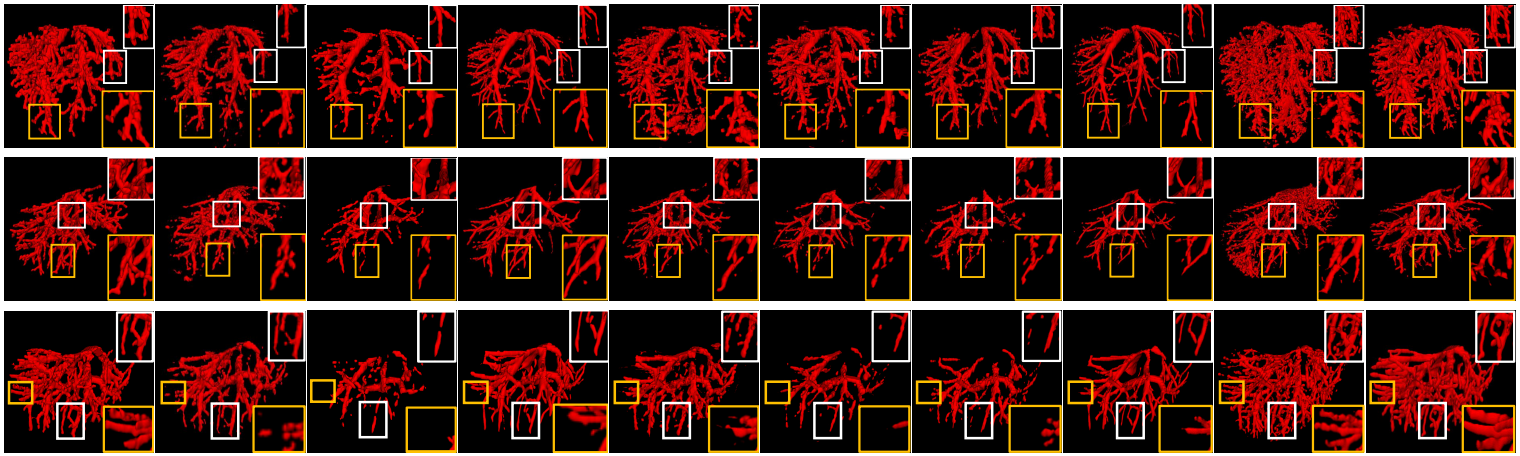
Quantitative evaluation on *3D-ircadb-01*. Tab. 3.2 provides the numerical evaluation of our proposed model, *HiDiff* [42], *MERIT* [54], *TransUNet* [55], *MedSegDiff* [39], *EnsemDiff* [38], *Swin UNETR* [2] and *nnUNet* [1]. We group the methods per network representation: 3D (using 3D convolutions) or 2.5D (using 2D convolutions

over 2.5D CT slices). The 3D non-diffusion methods such as *nnUNet* and *Swin UNETR* are comparable to or being outperformed by *MedSegDiff* and *HiDiff*, but are better than *EnsemDiff*. Intuitively, diffusion-based methods with dynamic conditioning like *MedSegDiff* and *HiDiff* perform better than the non-dynamic conditioning methods, like *EnsemDiff*. Although, *HiDiff*, *EnsemDiff* and *MedSegDiff* and our proposed model are built on diffusion models, our method still exceeds them in terms of *DSC* and *Sen* scores. Our model exceeds the best-performing baselines for individual metrics by 1.49% and 6.61% in *DSC* and *Sen* scores, as shown in Tab. 3.2. These improvements are due to the graph-attention conditioning, adding continuity and completeness to the vessel predictions. Moreover, our method has the lowest standard deviation for *DSC* and *Sen* compared to the other methods. This indicates that our model tends to make more stable predictions.

The *Spe* metric evaluates the degree of false positive predictions for a segmented liver vessel tree. *Swin UNETR*, *TransUNet*, *EnsemDiff*, *HiDiff* and *MedSegDiff* obtain comparable *Spe* scores. However, our method obtains a slightly lower score than the others. The lower *Spe* scores could be explained by the the missing small-vessel annotations. Interestingly, in Tab. 3.2, *MERIT* and *nnUNet* have the highest averaged *Spe* but with the lowest averaged *Sen*, which is a trade-off between segmentation accuracy and completeness.

Tab. 3.2 also reports vessel connectivity, in the *clDice* [52] and *Con* metrics. *Con* (in Eq. (3.11)) should ideally be as close as possible to 1. Additionally, we also report in the brackets the number of connected regions of the segmented vessel tree and the ground truth. Our method achieves the highest *clDice* score compared to the baselines, with the exception of *TransUNet*. The *clDice* scores of both our method and *TransUNet* are ≈75%, which indicates that our method can best fit the centerlines of the segmented liver vessel tree, following the ground truth. *nnUNet* achieves the third highest *clDice* score. However, the *clDice* calculation depends on the erosion centerline, which can introduce bias. *nnUNet* has the lowest *Sen* score, which indicates incomplete segmentations. The *Con* score of our method is the closest value to 1 compared to the other methods, which is indicative of the continuous predictions in our model. Interestingly, *MedSegDiff* which is also based on a diffusion model, obtains the worst connectivity score. This may be due to less precise auxiliary segmentations used in *MedSegDiff*.

Quantitative evaluation on *LiVS*. Fig. 3.5 (left) compares our method with *HiDiff* [42], *MERIT* [54], *MedSegDiff* [39], *TransUNet* [55] and *EnsemDiff* [38], *Swin UNETR* [2], *nnUNet* [1] and Gao et al. [24]. The ground truth of the liver vessels is discontinuous in the longitudinal direction, as shown in Fig. 3.6(j). On *LiVS* we cannot report *clDice* and *Con* scores, because of the discontinuous annotations. In



(a) Ours (b) *HiDiff* (c) *MERIT* (d) *TransUNet* (e) *MedSegDiff* (f) *EnsemDiff* (g) *SwinUNETR* (h) *nnUNet* (i) *Gao et al.* (j) Ground truth

Figure 3.4: **Visualizations on the *3D-ircadb-01* [23] dataset.** (a) The liver vessel tree segmented by our proposed model; (b) – (i) are the segmentations from baselines: *HiDiff* [42], *MERIT* [54], *TransUNet* [55], *MedSegDiff* [39], *EnsemDiff* [38], *Swin UNETR* [2], *nnUNet* [1], and [24]; (j) The ground truth. Yellow/white boxes highlight segmentation completeness and continuity.

Table 3.2: **Results on the *3D-ircadb-01* [23] dataset**. We compare our model with: *HiDiff*[42], *MERIT*[54], *TransUNet*[55], *MedSegDiff*[39], *EnsemDiff*[38], *Swin UNETR* [2], *nnUNet* [1] and [24]. Our method is the best in terms of *DSC*, *clDice*, *Sen* and *Con* scores, but worse in *Spe* scores. Interestingly, *nnUNet* has the highest *Spe* score and the lowest *Sen* score, which may be due to a trade-off between detailed and accurate segmentation.

	Type	*DSC* (%)	*clDice* (%)	*Sen* (%)	*Spe* (%)	*Con* ($\rightarrow 1$)
Gao et al.[24]	2D	60.19 ± 4.69	56.48 ± 11.74	64.98 ± 4.48	99.80 ± 0.10	8.73 (96/11)
nnUNet[1]	3D	60.54 ± 6.86	71.60 ± 4.37	44.76 ± 8.09	**99.99** ± 0.00	2.36 (26/11)
Swin UNETR[2]	3D	55.61 ± 9.80	62.75 ± 7.91	42.30 ± 11.46	99.98 ± 0.01	5.45 (60/11)
TransUNet[55]	3D	69.77 ± 3.73	**75.83** ± 5.50	58.63 ± 11.70	99.97 ± 0.01	3.73 (41/11)
EnsemDiff[38]	2.5D	55.07 ± 9.70	60.99 ± 9.64	40.45 ± 10.60	99.98 ± 0.02	7.18 (79/11)
MedSegDiff[39]	2.5D	59.67 ± 7.74	66.02 ± 7.87	47.42 ± 10.43	99.95 ± 0.05	8.64 (95/11)
MERIT[54]	2.5D	49.50 ± 9.57	53.00 ± 8.79	34.30 ± 8.81	99.99 ± 0.01	5.50 (59/11)
HiDiff[42]	2.5D	63.32 ± 4.01	60.24 ± 2.71	54.45 ± 7.25	99.94 ± 0.01	6.70 (72/11)
Ours	2.5D	**71.26** ± 1.93	74.61 ± 1.21	**71.59** ± 4.07	99.89 ± 0.04	**1.09** (12/11)

Fig. 3.5, our method has the best *DSC* and *Sen* scores. Especially in terms of *Sen*, our method performs better than the baselines, indicating a more complete vessel segmentation. In terms of *DSC* scores, we are comparable to *nnUNet* because the *DSC* scores are susceptible to outliers, as shown in Fig. 3.5 (right). The methods of *HiDiff* and Gao *et al.* have comparable *Sen* scores to *nnUNet*, but its *DSC* score is lower, because of having the lowest *Spe* score of all methods. *SwinUNetr* has the lowest *DSC* scores and *MERIT* has the lowest *Sen* scores. As with the *3D-ircadb-01* dataset, our improvements are due to the graph-attention conditioning. Although *HiDiff* and *MedSegDiff* have lower scores than our model in terms of *DSC* and *Sen*, it outperforms the *EnsemDiff* model. This may be due to the dynamic conditioning method used in *HiDiff* and *MedSegDiff*. The standard deviations of our predictions and predictions of *HiDiff* and *MedSegDiff* are comparable, and they are both lower than the standard deviation of *EnsemDiff*. This could indicate that the dynamic conditioning diffusion model (ours, *HiDiff* and *MedSegDiff*) can provide more stable results than the vanilla conditioning model (*EnsemDiff*).

However, these diffusion-based methods are less efficient than deterministic methods during inference, as shown in the Fig. 3.5. Our model is slower than *MedSegDiff* and *EnsemDiff* when the inference batch size is set to 50 slices. This slowdown is due to the additional cost of graph attention and local feature integration (LFI) in our model. The graph attention introduces anatomic structure into the model, improving accuracy, but also increases inference time compared to other diffusion baselines. Similarly, LFI compensates for the sparse node representation, but the grid sample layer used in LFI results in heavier computation.

	Repr. type	DSC (%)	Sen (%)	Spe (%)	*sec. /slice*	steps
Gao et al.[24]	2D	73.14 ± 11.44	76.84 ± 11.57	99.63 ± 0.34	< 0.1	-
nnUNet[1]	3D	81.39 ± 5.04	76.06 ± 7.48	99.91 ± 0.05	< 0.1	-
Swin UNETR[2]	3D	65.54 ± 6.65	62.71 ± 10.31	99.76 ± 0.13	< 0.1	-
TransUNet[55]	3D	78.21 ± 6.17	75.11 ± 7.58	99.86 ± 0.08	< 0.1	-
EnsemDiff[38]	2.5D	70.00 ± 9.20	56.27 ± 10.81	**99.97** ± 0.03	20.67	1000
MedSegDiff[39]	2.5D	76.85 ± 7.03	65.96 ± 9.43	99.96 ± 0.04	26.92	1000
MERIT[54]	2.5D	69.54 ± 6.72	56.63 ± 8.43	99.95 ± 0.04	< 0.1	-
HiDiff[42]	2.5D	70.53 ± 6.66	78.32 ± 5.49	99.63 ± 0.12	0.25	10
Ours	2.5D	**81.41** ± 6.64	**81.35** ± 6.93	99.84 ± 0.12	43.21	1000

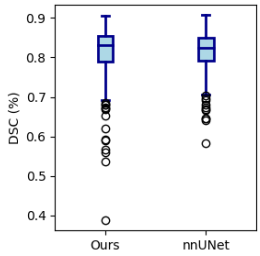


Figure 3.5: **Results on the LiVS [24] dataset.** *Left:* The performance of our method compared with *HiDiff* [42], *MERIT* [54], *TransUNet* [55], *MedSegDiff* [39], *EnsemDiff* [38], *Swin UNETR* [2], *nnUNet* [1] and Gao et al. [24]. *clDice* and *Con* are not applicable for the *LiVS* dataset with discontinuous annotations. Our method outperforms others in terms of *Sen* scores. *Spe* of our model is slightly lower than *EnsemDiff*, which is a trade-off between completeness (*Sen*) and accuracy (*Spe*) of vessel segmentation. *Right:* Our *DSC* score is comparable with *nnUNet* because high-contrast tumors in the liver cause more outliers (shown in the box plot), negatively affecting the averaged *DSC* score. Deterministic methods are more efficient than generative methods in inference.

3.5.6 Qualitative comparison

Qualitative evaluation on *3D-ircadb-01*. In Fig. 3.4 we provide a qualitative evaluation on three test cases from the *3D-ircadb-01* dataset. Compared to the other methods, the appearance of our predicted vessel-tree segmentation (first column) is the most similar to the ground truth (last column). The vascular structures marked by the yellow and white boxes in Fig. 3.4 are almost completely segmented by our method, while the other methods oversegment or miss parts of the vessel tree. Especially the segmentation results of *MERIT* [54], *Swin UNETR* [2] and *nnUNet* [1], are visibly sparse. These results relate to the low *DSC* and *Sen* scores of *MERIT*, *Swin UNETR* and *nnUNet* in Tab. 3.2. Although the vessel structures of *EnsemDiff* [38] (sixth column) and *MedSegDiff* [39] (fifth column) are denser than those of the non-diffusion methods, they provide more discontinuous vessel masks at the extremities of the tree (*i.e.* for smaller vessels).

In Fig. 3.4 we also highlight the connectivity of the distal liver vessel branches in the yellow boxes. The visualizations show that our segmentation is as continuous as the ground truth. This corresponds to a *Con* score closer to 1 in Tab. 3.2. Comparing the vessel branches of *nnUNet* with the other methods for the distal vessels, we see that these are precise, being exceeded only by our method. This is again consistent with the *Con* score of *nnUNet* in Tab. 3.2 – the second best score.

Qualitative evaluation on *LiVS*. In Fig. 3.6, we provide a qualitative comparison in 3D

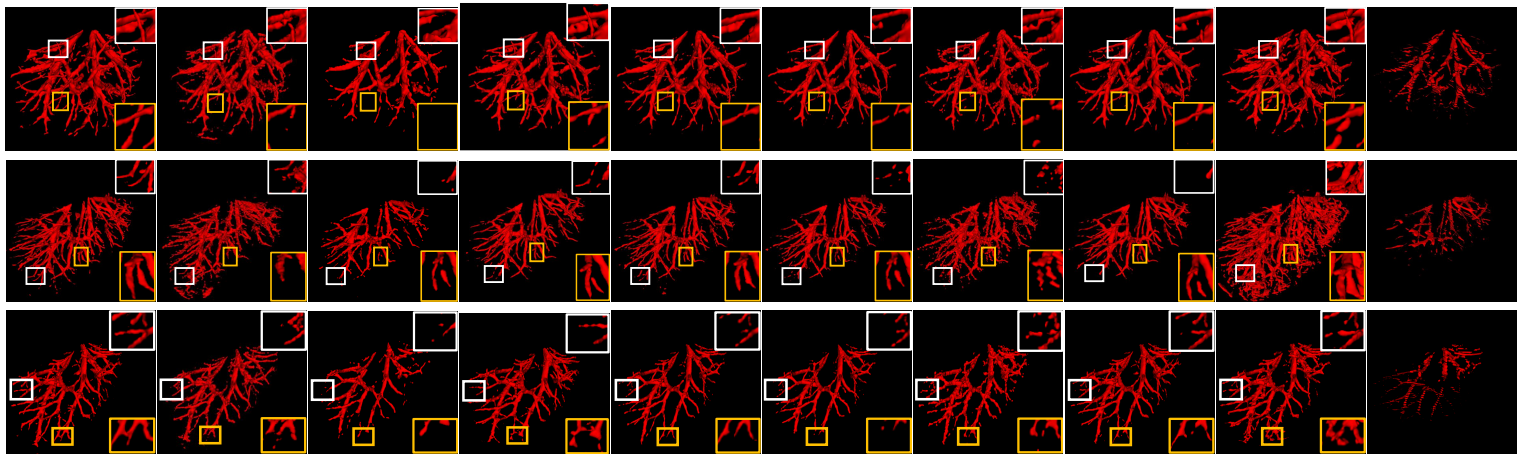
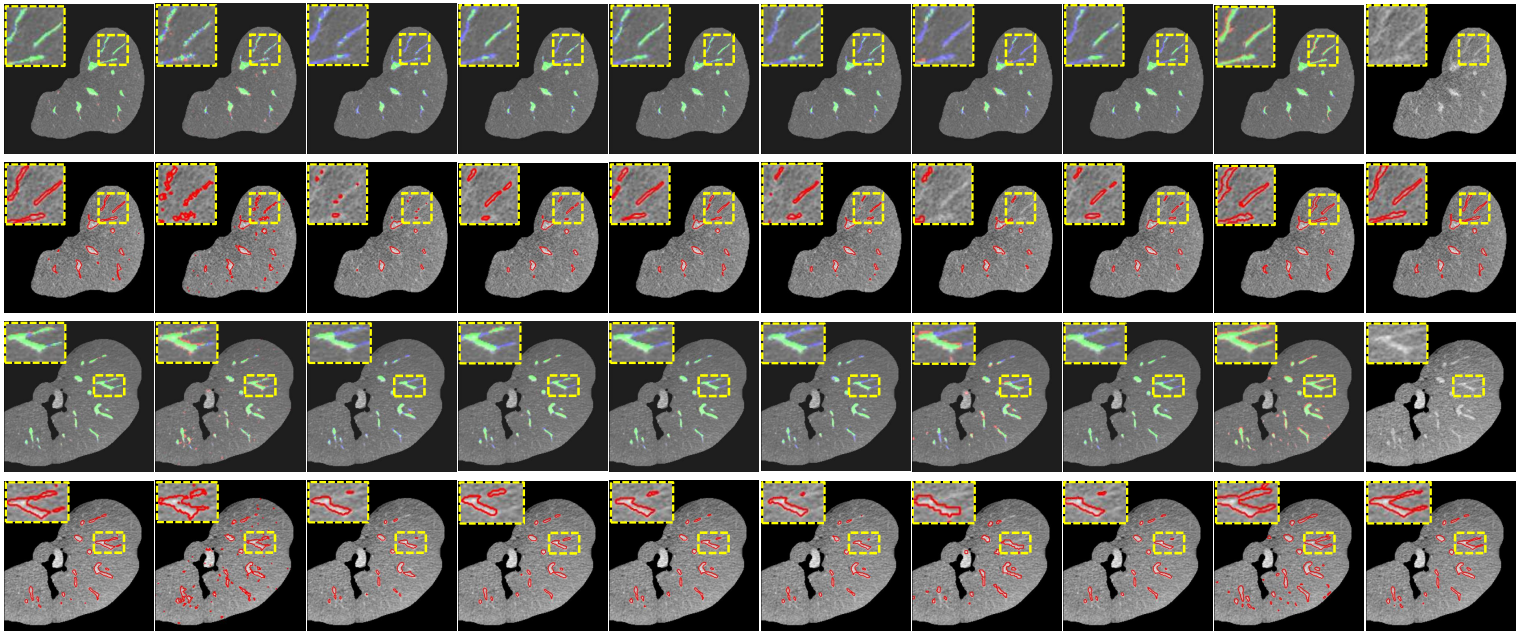
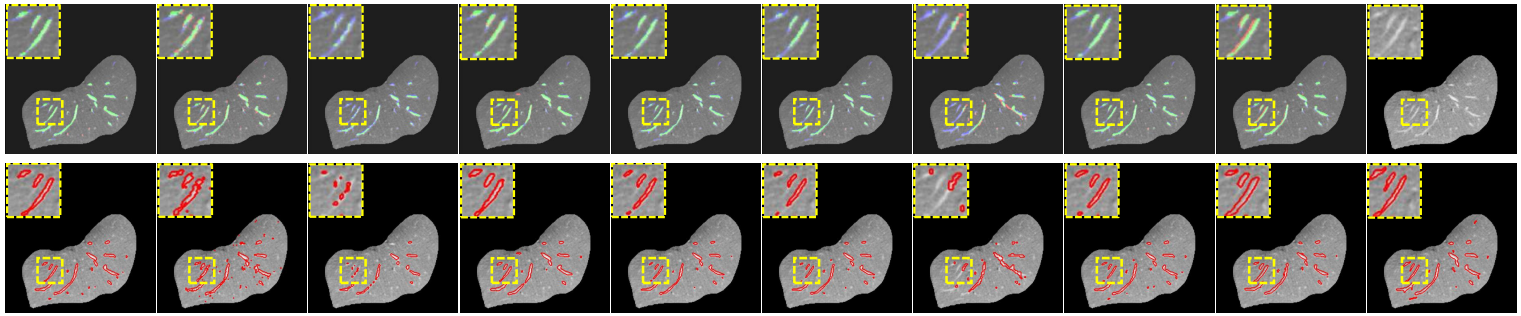


Figure 3.6: **Visualizations of the *LiVS* [24] dataset.** (a) The liver vessel tree segmented by our proposed model; (b)–(i) are the liver vessel tree segmented by the baselines: *HiDiff* [42], *MERIT* [54], *TransUNet* [55], *MedSegDiff* [39], *EnsemDiff* [38], *Swin UNETR* [2], *nnUNet* [1] and Gao et al. [24]; (j) The discontinuous (partially annotated) ground truth liver vessel tree. Yellow and white boxes show completeness and continuity between our model and baselines. Enlarged views shown in the corners.



(a) Ours | (b) *HiDiff* | (c) *MERIT* | (d) *TransUNet* | (e) *MedSegDiff* | (f) *EnsemDiff* | (g) *SwinUNetr* | (h) *nnUNet* | (i) *Gao et al.* | (j) *Ground truth*

(Part 1/2) Cross-sectional visualization examples on *LiVS* dataset (first two cases).



(a) Ours | (b) *HiDiff* | (c) *MERIT* | (d) *TransUNet* | (e) *MedSegDiff* | (f) *EnsemDiff* | (g) *SwinUNetr* | (h) *nnUNet* | (i) *Gao et al.* | (j) *Ground truth*

Figure 3.7: **(Part 2/2) Cross-sectional visualizations on *LiVS* [24] dataset.** We compare segmentation masks from our proposed model with: *HiDiff* [42], *MERIT* [54], *TransUNet* [55], *MedSegDiff* [39], *EnsemDiff* [38], *Swin UNETR* [2], *nnUNet* [1] and Gao et al. [24]. The first and second rows are respectively overlaid masks and contour comparisons. Green indicates true positives, red false positives, and blue false negatives. Differences are highlighted in yellow boxes. Our method better preserves fine vessel connectivity.

Table 3.3: **Ablation study on *3D-ircadb-01* dataset.** We ablate the effect of each individual component presented in Fig. 3.3 and the effect of the post-processing: (A) vanilla conditioning model [38]; (B) dynamic conditioning model; (C) multiscale graph-attention conditioning; and (D) post-processing. Interestingly, the *(A) vanilla model* and the *(A) vanilla model* combined with *(B) dynamic conditioning* perform best in terms of *Spe* scores. This is because predicting less structures (low *Sen* scores) entails fewer false positives. However, the overall combination performs best in *DSC* score.

(A) Vanilla conditioning	(B) Dynamic conditioning	(C) Dynamic multiscale graph-attention	(D) Post-processing	DSC(%)	Sen(%)	Spe(%)
✓	×	×	×	55.07 ± 9.70	40.45 ± 10.6	99.98 ± 0.02
✓	✓	×	×	59.00 ± 5.43	44.22 ± 6.19	**99.98** ± 0.01
✓	✓	✓	×	62.69 ± 3.93	**79.80** ± 6.33	99.70 ± 0.17
✓	✓	✓	✓	**71.26** ± 1.93	71.59 ± 4.07	99.89 ± 0.04

on three test cases between our model and the eight baselines. Although it is hard to directly compare the segmentations of the different methods, due to the discontinuous ground truth (shown in Fig. 3.6(j)), inter-method comparison can still be informative. The yellow and white boxes in Fig. 3.6 show that most baselines predict fractured small vessels for the distal branch, while our segmented vessel structures are denser and more complete.

We compare vessel completeness in Fig. 3.7, where we show a qualitative comparison for the slices of the three cases in Fig. 3.6, in a 2D cross-sectional view. The first, third and fifth rows show the overlayed segmention masks of the liver vessel mask of different methods, when compared to the ground truth mask. Green, red and blue colors represent true positive, false positive and false negative, respectively. The blue areas of the overlaid masks from columns (b) to (i) reflect the missing contours of the baselines. Our model has fewer blue regions, indicating that our segmentation is more complete for both the large and small vessel structures. The red contours in the second, forth and final rows of Fig. 3.7 outline the boundary of the predicted liver vessel segmentation of different methods. The contours in the last column (j) correspond to the boundary of ground truth vessel mask. We enlarged the regions using the yellow boxes, to highlight differences in the vessel completeness. Comparing the contours of the baselines from column (b) to (i) with the ground truth, tiny vessel blobs are not outlined. For the baseline (i), the contours reflect a heavy oversegmentation. Our model benefits from the long-range feature dependency encoded in the multi-scale graph-attention conditioning.

3.5.7 Model ablation study

Effect of different model components and post-processing. For completeness, we perform the ablation studies on both the *3D-ircadb-01* and *LiVS* datasets. Because the

Table 3.4: **Ablation study on *LiVS* dataset.** We ablate the effect of each individual component presented in Fig. 3.3 and the effect of the post-processing: (A) vanilla conditioning model [38]; (B) dynamic conditioning model; (C) multiscale graph-attention conditioning; and (D) post-processing. Inference ensembling is used due to the discontinuous annotations of the dataset. Post-processing is not mandatory for this kind of dataset.

(A) Vanilla conditioning	(B) Dynamic conditioning	(C) Dynamic multiscale graph-attention	(D) Post-processing	DSC(%)	Sen(%)	Spe(%)
✓	×	×	×	70.00 ± 9.20	56.27 ± 10.81	**99.97** ± 0.03
✓	✓	×	×	73.39 ± 7.55	60.96 ± 9.51	99.96 ± 0.03
✓	✓	✓	×	**81.41** ± 6.64	**81.35** ± 6.93	99.84 ± 0.12
✓	✓	✓	✓	81.04 ± 6.50	78.02 ± 7.19	99.88 ± 0.11

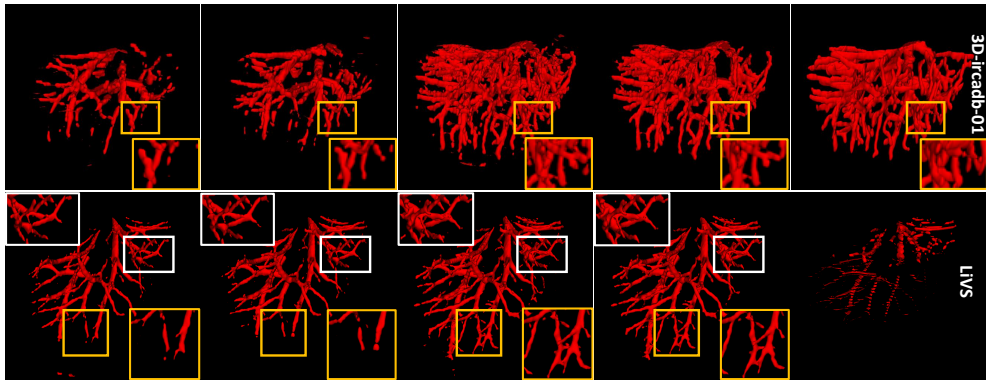


Figure 3.8: **Visualization of the per-component ablation study for the *3D-IRCADb-01* dataset in Tab. 3.3 and the *LiVS* dataset in Tab. 3.4.** Each column shows the cumulative effect of adding components (A) to (D), compared against the ground truth (E). (A) Vanilla conditioning; (B) Dynamic conditioning; (C) Dynamic multiscale graph-attention; (D) Post-processing; (E) Ground truth.

three components of our model : (i) vanilla conditioning, (ii) dynamic conditioning and (iii) multiscale graph-attention, are highly interconnected (see Fig. 3.3), we cannot evaluate them independently. To explore the influence of different conditioning levels on the performance of liver vessel segmentation, we perform an additive ablation study in Tab. 3.3 and Tab. 3.4, and visualize the ablation in Fig. 3.8. We start from the *(A) vanilla conditioning model* and subsequently add new conditioning components to it. Noteworthy, the *(A) vanilla model* (in the first row) and the combination of the vanilla model with the *(B) dynamic conditioning model* (in the second row) achieve the highest *Spe* scores. This is due to these models predicting fewer structures in the segmentation masks (as shown by the low *Sen* score and in Fig. 3.8), and therefore

Table 3.5: **Ablation study testing the effect of the number of graph nodes, on the *3D-IRCADb-01* and *LiVS* datasets.** No post-processing or inference ensembling was used in this ablation study on node number. As the number of graph nodes increase, the computations (GFLOPs) increase. Using $32^2 \times 3$ provides a good tradeoff betwen accuracy and computations, while balancing *Sen* and *Spe*.

#nodes	*3D-ircadb-01*			*LiVS*			GFLOPs
per batch	DSC(%)	Sen(%)	Spe(%)	DSC(%)	Sen(%)	Spe(%)	per batch
$16^2 \times 3$	61.69 ± 4.03	76.99 ± 7.95	99.69 ± 0.22	66.13 ± 10.46	88.02 ± 4.32	99.29 ± 0.29	533.95
$32^2 \times 3$	62.69 ± 3.93	79.80 ± 6.33	99.70 ± 0.17	71.90 ± 8.85	87.07 ± 4.71	99.51 ± 0.20	565.20
$64^2 \times 3$	62.71 ± 4.88	54.80 ± 9.00	99.92 ± 0.04	75.57 ± 5.72	67.87 ± 8.05	99.90 ± 0.05	690.17

having fewer false positives. The final model combining all three components: *(A) vanilla conditioning*, *(B) dynamic conditioning* and *(C) multiscale graph-attention* has the highest *DSC*, and *Sen* scores. The graph-attention conditioning contributes considerably to the performance of our model. Post-processing is an effective way to remove noisy predictions, and to improve the *DSC* score as shown in Tab. 3.3, but it can negatively affect the *Sen*. However, for datasets with discontinuous annotations (such as *LiVS*), post-processing is not mandatory because inference ensembles can also remove noisy predictions. Overall, we conclude that all components have a beneficial effect on the vessel segmentation scores.

The effect of different graph node numbers. We perform ablation studies testing the effect of the number of graph nodes, on *3D-ircad-01* and *LiVS* datasets. To isolate the effect of the number of graph nodes, we do not apply post-processing or inference ensembling, in this ablation study. In addition to the node numbers of 32×32×3 per batch used in our implementation, we also report results for configurations of 16×16×3 and 64×64×3, as shown in Tab. 3.5. Although our model with a configuration of 64×64×3 achieves the highest *DSC* and *Spe* scores on both datasets, it yields the lowest *Sen* score and incurs higher computational complexity in terms of GFLOPs. The lower *Sen* score of our model with the 64×64×3 configuration indicates relatively incomplete vessel segmentation, making it insufficient to meet the requirement for segmentation completeness. The sparser node configuration of 16×16×3 results in a lower *Spe* score, indicating a higher number of false positives in the segmentation. To balance *Sen* and *Spe* while considering computational complexity, the 32×32×3 configuration used in our implementation represents a good tradeoff.

Graph-attention in inference. In inference, we input a fully-connected graph in the component (C), as shown in Fig. 3.2. While the node coordinates are uniformly distributed in the graph, the edge weights between nodes are adapted by the trained graph-attention layer. Fig. 3.9 visualizes the learned edge weights for an input fully-connected graph. In Fig. 3.9, the cross-sectional view shows that the vessel area and

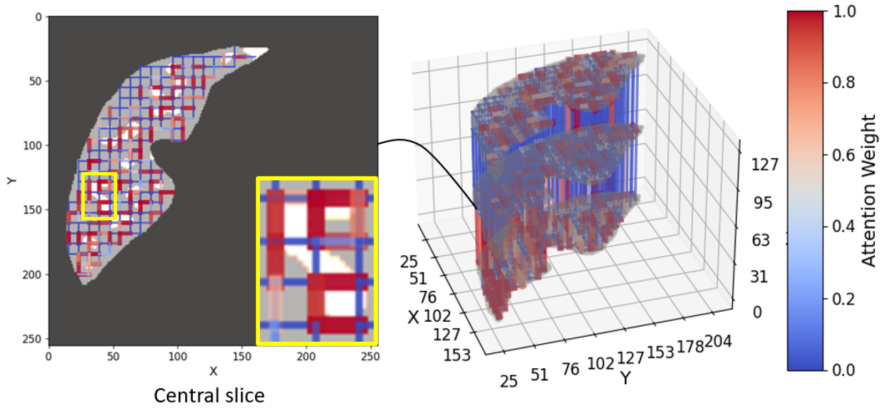


Figure 3.9: **Edge attention weights of a fully-connected graph in inference.** Blue/red represent low/high attention weights, respectively. We also show the box enlarged in the bottom right corner. The vessel area and its neighborhood attract more edge attention, thus demonstrating the utility of inputting a fully-connected graph in inference.

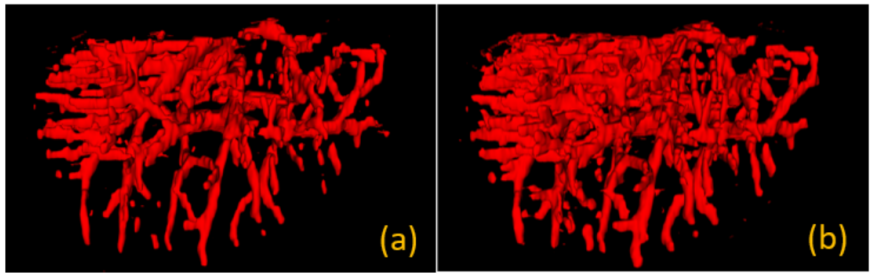


Figure 3.10: **Inference difference between using an empty graph, without edges (a) and a fully-connected graph (b) as the input.** A fully-connected graph leads to more dense and continuous predictions.

its neighborhood have higher-magnitude edge-attention. Additionally, in Fig. 3.10 we compare the difference in predictions when inputting an empty graph, without edges (a) and a fully-connected graph (b). Using an empty graph leads to sparser predictions. This analysis demonstrates that graph attention layers are still effective, even with a uniform graph as input.

3.6 Discussion

3.6.1 Limitation

Fig. 3.11 shows the limitation of our model and contains examples with the worst *DSC* scores for the *LiVS* and *3D-ircadb-01* datasets. On these worst cases, our

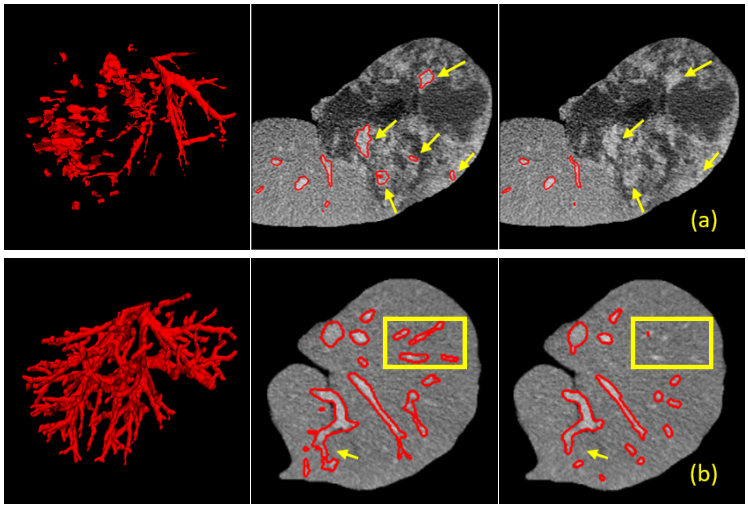


Figure 3.11: **Analysis of model limitations.** (a) First row: Visualization of a failure case on the *LiVS* dataset. (b) Second row: Visualization of a failure case on the *3D-ircadb-01* dataset. In case (a) the failure is due to the contrast marked by yellow arrows around the tumors being misclassified as vessel structures. In case (b) the failure is due to the unlabeled vessels [14, 26] (highlighted in the yellow box), causing low *DSC* scores. The model also makes a mistake by predicting segmentation masks where there should not be (over-segmentation), as shown by the yellow arrow. This is caused by the inconsistency in annotations in the *3D-ircadb-01* dataset [23].

proposed model inaccurately predicts vessel segmentations when there is a large tumor surrounded by contrast-rich regions, as seen in Fig. 3.11(a). This failure is reflected by the outliers of the *DSC* scores of the *LiVS* dataset in Fig. 3.5. All tumors in *3D-ircadb-01* are characterized as low-intensity regions, and are not surrounded by contrast, so this failure appears only on the *LiVS* dataset. Fig. 3.11(b) shows a worst performing example on the *3D-ircadb-01* dataset. The low *DSC* score is due to missing annotations [14, 26]. The yellow box highlights a case where vessels were not annotated. Our model correctly predicts this, while still being penalized in the *DSC* scores. Using yellow arrows we indicate regions where our model over-predicts structure that is not truly present. The inconsistency in the annotation quality in the *3D-ircadb-01* dataset[23] is an additional challenge in the training of the model. This may lead to that the model learns to focus on the wrong information when predicting segmentations.

3.7 Conclusion

In this study, we focus on liver vessel segmentation from CT volumes. To this end, we propose to augment conditional diffusion models with geometric graph-structure computed at multiple resolutions. The role of the graph structure is to ensure connectivity in the segmentation, across neighboring slices in the CT volume. Moreover, we use multi-scale features in the graph, thus allowing the model to focus on small vessels, that otherwise would be missed. Our proposed model achieves state-of-the-art results, in terms of *DSC*, *Sen* and vessel connectivity on two standard benchmarks: *3D-ircadb-01* [23] and *LiVS* [24], when compared to baselines such as *HiDiff* [42], *MERIT*[54], *TransUNet* [55], *MedSegDiff* [39], *EnsemDiff* [38], *Swin UNETR* [2], *nnUNet* [1] and Gao et al. [24].

References

[1] F. Isensee, P. F. Jaeger, S. A. Kohl, et al. “nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation”. In: *Nature methods* 18.2 (2021), pages 203–211.

[2] A. Hatamizadeh, V. Nath, Y. Tang, et al. “Swin UNETR: Swin transformers for semantic segmentation of brain tumors in MRI images”. In: *International MICCAI Brainlesion Workshop*. Sept. 2021, pages 272–284.

[3] X. Li, P. Ramadori, D. Pfister, et al. “The immunological and metabolic landscape in primary and metastatic liver cancer”. In: *Nature Reviews Cancer* 21.9 (2021), pages 541–557.

[4] G. Disibio and S. W. French. “Metastatic patterns of cancers: results from a large autopsy study”. In: *Archives of pathology & laboratory medicine* 132.6 (2008), pages 931–939.

[5] J. Llovet, R. Kelley, and A. Villanueva. “Hepatocellular carcinoma”. In: *Nat Rev Dis Primers* 7.6 (2021).

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

[7] H. Huang. “Influence of blood vessel on the thermal lesion formation during radiofrequency ablation for liver tumors”. In: *Medical physics* 40.7 (2013), page 073303.

[8] S. Survarachakan, E. Pelanis, Z. Khan, et al. “Effects of enhancement on deep learning based hepatic vessel segmentation”. In: *Electronics* 10.10 (2021), page 1165.

[9] J. Lamy, O. Merveille, B. Kerautret, and N. Passat. “A Benchmark Framework for Multi-region Analysis of Vesselness Filters”. In: *IEEE Transactions on Medical Imaging* 41.12 (2022), pages 3649–3662.

[10] Y. Cheng, X. Hu, J. Wang, et al. “Accurate vessel segmentation with constrained b-snake”. In: *IEEE Transactions on Image Processing* 24.8 (2015), pages 2440–2455.

[11] M. Chung, J. Lee, J. W. Chung, and Y.-G. Shin. “Accurate liver vessel segmentation via active contour model with dense vessel candidates”. In: *Computer methods and programs in biomedicine* 166 (2018), pages 61–75.

[12] O. Friman, M. Hindennach, C. Kühnel, and H. Peitgen. “Multiple hypothesis template tracking of small 3D vessel structures”. In: *Medical image analysis* 14.2 (2010), pages 160–171.

[13] S. Cetin and G. Unal. “A higher-order tensor vessel tractography for segmentation of vascular structures”. In: *IEEE Transactions on Medical Imaging* 34.10 (2015), pages 2172–2185.

[14] Q. Huang, J. Sun, H. Ding, et al. “Robust liver vessel extraction using 3D U-Net with variant dice loss function”. In: *Computers in Biology and Medicine* 101 (2018), pages 153–162.

[15] T. Kitrungrotsakul, X. Han, Y. Iwamoto, et al. “VesselNet: A deep convolutional neural network with multi pathways for robust hepatic vessel segmentation”. In: *Computerized Medical Imaging and Graphics* 75 (2019), pages 74–83.

[16] M. Wu, Y. Qian, X. Liao, et al. “Hepatic vessel segmentation based on 3D swin-transformer with inductive biased multi-head self-attention”. In: *BMC Medical Imaging* 23.1 (2023), pages 1–14.

[17] R. Li, Y. Huang, H. Chen, et al. “3D graph-connectivity constrained network for hepatic vessel segmentation”. In: *IEEE Journal of Biomedical and Health Informatics* 26.3 (2021), pages 1251–1262.

[18] B. Nichyporuk, J. Cardinell, J. Szeto, et al. “Rethinking generalization: The impact of annotation style on medical image segmentation”. In: *Journal of Machine Learning for Biomedical Imaging* (2022).

[19] J. Ho, A. Jain, and P. Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems*. Volume 33. 2020, pages 6840–6851.

[20] A. Nichol and P. Dhariwal. “Improved denoising diffusion probabilistic models”. In: *International Conference on Machine Learning*. PMLR, July 2021, pages 8162–8171.

[21] S. Brody, U. Alon, and E. Yahav. “How Attentive are Graph Attention Networks?” In: *International Conference on Learning Representations*. 2022.

[22] Y. Chen, S. Liu, and X. Wang. “Learning continuous image representation with local implicit image function”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pages 8628–8638.

[23] L. Soler, A. Hostettler, V. Agnus, et al. “3D image reconstruction for comparison of algorithm database: A patient specific anatomical and medical image database”. In: ().

[24] Z. Gao, Q. Zong, Y. Wang, et al. “Laplacian salience-gated feature pyramid network for accurate liver vessel segmentation”. In: *IEEE Transactions on Medical Imaging* (May 2023).

[25] B. Ibragimov, D. Toesca, D. Chang, et al. “Combining deep learning with anatomical analysis for segmentation of the portal vein for liver SBRT planning”. In: *Physics in Medicine & Biology* 62.23 (2017), page 8943.

[26] Q. Yan, B. Wang, W. Zhang, et al. “Attention-guided deep neural network with multi-scale feature fusion for liver vessel segmentation”. In: *IEEE Journal of Biomedical and Health Informatics* 25.7 (2020), pages 2629–2642.

[27] X. Wang, X. Zhang, G. Wang, et al. “TransFusionNet: Semantic and Spatial Features Fusion Framework for Liver Tumor and Vessel Segmentation Under JetsonTX2”. In: *IEEE Journal of Biomedical and Health Informatics* 27.3 (2022), pages 1173–1184.

[28] D. Zhang, S. Liu, S. Chaganti, et al. “Graph attention network based pruning for reconstructing 3D liver vessel morphology from contrasted CT images”. In: *CoRR* (2020).

[29] O. Ronneberger, P. Fischer, and T. Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention*. Springer International Publishing, 2015, pages 234–241.

[30] R. Zhao, B. Qian, X. Zhang, et al. “Rethinking dice loss for medical image segmentation”. In: *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2020, pages 851–860.

[31] Y. Liu, J. Yu, and Y. Han. “Understanding the effective receptive field in semantic image segmentation”. In: *Multimedia Tools and Applications* 77 (2018), pages 22159–22171.

[32] 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.

[33] P. Velikovi, G. Cucurull, A. Casanova, et al. “Graph Attention Networks”. In: *International Conference on Learning Representations* (2018).

[34] X. Zhang, Q. Zhu, T. Hu, et al. “Joint high-resolution feature learning and vessel-shape aware convolutions for efficient vessel segmentation”. In: *Computers in Biology and Medicine* 191 (2025), page 109982.

[35] D. Dai, C. Dong, S. Xu, et al. “Ms RED: A novel multi-scale residual encoding and decoding network for skin lesion segmentation”. In: *Medical image analysis* 75 (2022), page 102293.

[36] Q. Yan, S. Liu, S. Xu, et al. “3D medical image segmentation using parallel transformers”. In: *Pattern Recognition* 138 (2023), page 109432.

[37] D. Dai, C. Dong, Q. Yan, et al. “I2u-net: A dual-path u-net with rich information interaction for medical image segmentation”. In: *Medical Image Analysis* 97 (2024), page 103241.

[38] J. Wolleb, R. Sandkühler, F. Bieder, et al. “Diffusion models for implicit image segmentation ensembles”. In: *International Conference on Medical Imaging with Deep Learning*. PMLR, Dec. 2022, pages 1336–1348.

[39] J. Wu, R. FU, H. Fang, et al. “MedSegDiff: Medical Image Segmentation with Diffusion Probabilistic Model”. In: *Medical Imaging with Deep Learning*. 2023.

[40] Z. Xing, L. Wan, H. Fu, et al. “Diff-UNet: A Diffusion Embedded Network for Volumetric Segmentation”. In: *CoRR* (2023).

[41] F. Bieder, J. Wolleb, A. Durrer, et al. “Memory-Efficient 3D Denoising Diffusion Models for Medical Image Processing”. In: *Medical Imaging with Deep Learning*. 2023.

[42] T. Chen, C. Wang, Z. Chen, et al. “HiDiff: hybrid diffusion framework for medical image segmentation”. In: *IEEE Transactions on Medical Imaging* (2024).

[43] R. Selvan, T. Kipf, M. Welling, et al. “Graph refinement based airway extraction using mean-field networks and graph neural networks”. In: *Medical image analysis* 64 (2020), page 101751.

[44] S. Shin, S. Lee, I. Yun, and K. Lee. “Deep vessel segmentation by learning graphical connectivity”. In: *Medical Image Analysis* 58 (2019), page 101556.

[45] Z. Tian, X. Li, Y. Zheng, et al. “Graph-convolutional-network-based interactive prostate segmentation in MR images”. In: *Medical physics* 47.9 (2020), pages 4164–4176.

[46] R. Soberanis-Mukul, N. Navab, and S. Albarqouni. “Uncertainty-based graph convolutional networks for organ segmentation refinement”. In: *Medical Imaging with Deep Learning*. PMLR. 2020, pages 755–769.

[47] L. Liu, J. Wolterink, C. Brune, and R. Veldhuis. “Anatomy-aided deep learning for medical image segmentation: a review”. In: *Physics in Medicine & Biology* 66.11 (2021), 11TR01.

[48] M. Jaderberg, K. Simonyan, A. Zisserman, et al. “Spatial transformer networks”. In: *Advances in neural information processing systems* 28 (2015).

[49] P. Yushkevich, J. Piven, H. Hazlett, et al. “User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability”. In: *Neuroimage* 31.3 (July 2006), pages 1116–1128.

[50] J. Weaver. “Centrosymmetric (cross-symmetric) matrices, their basic properties, eigenvalues, and eigenvectors”. In: *The American Mathematical Monthly* 92.10 (1985), pages 711–717.

[51] D. Müller, I. Soto-Rey, and F. Kramer. “Towards a guideline for evaluation metrics in medical image segmentation”. In: *BMC Research Notes* 15.1 (2022), page 210.

[52] S. Shit, J. Paetzold, A. Sekuboyina, et al. “clDice-a novel topology-preserving loss function for tubular structure segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pages 16560–16569.

[53] M. Gegúndez-Arias, A. Aquino, J. Bravo, and D. Marín. “A function for quality evaluation of retinal vessel segmentations”. In: *IEEE Transactions on Medical Imaging* 31.2 (2011), pages 231–239.

[54] M. M. Rahman and R. Marculescu. “Multi-scale hierarchical vision transformer with cascaded attention decoding for medical image segmentation”. In: *Medical Imaging with Deep Learning*. PMLR. 2024, pages 1526–1544.

[55] J. Chen, J. Mei, X. Li, et al. “TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers”. In: *Medical Image Analysis* 97 (2024), page 103280.