Structure-wise Uncertainty for Curvilinear Image Segmentation

Saumya Gupta SAUMGUPTA@CS.STONYBROOK.EDU
Department of Computer Science,
Stony Brook University, NY, USA

Xiaoling Hu XIHU3@MGH.HARVARD.EDU
Athinoula A. Martinos Center for Biomedical Imaging,
Massachusetts General Hospital and Harvard Medical School, MA, USA

Chao Chen CHAO.CHEN.1@STONYBROOK.EDU
Department of Biomedical Informatics,
Stony Brook University, NY, USA

Abstract

Segmenting curvilinear structures like blood vessels and roads poses significant challenges due to their intricate geometry and weak signals. To expedite large scale annotation, it is essential to adopt semi-automatic methods such as proofreading by human experts. In this abstract, we focus on estimating uncertainty for such tasks, so that highly uncertain, and thus error-prone structures can be identified for human annotators to verify. Unlike prior work that generate pixel-wise uncertainty maps, we believe it is essential to measure uncertainty in the units of topological structures, e.g., small pieces of connections and branches. To realize this, we employ tools from topological data analysis, specifically discrete Morse theory (DMT), to first extract the structures and then reason about their uncertainties. On multiple 2D and 3D datasets, our methodology generates superior structure-wise uncertainty maps compared to existing models. Code available at https://github.com/Saumya-Gupta-26/struct-uncertainty

Keywords: Topological Representation, Discrete Morse Theory, Structural Uncertainty, Image Segmentation, Curvilinear Structures

1. Introduction

Curvilinear segmentation is an essential initial step in various medical and non-medical applications, involving the precise extraction of fine-scale structures, such as blood vessels, nerves, and other elongated objects (González-Hidalgo, 2016; Kv et al., 2023). For example, extraction of retinal vasculature is an essential precursor to understanding disease progression and assessing therapeutic effects (Fraz et al., 2012). In civil engineering, road network and railway track segmentation can support urban planning and transportation system optimization (Mnih and Hinton, 2010). Despite the success of deep learning (Chen et al., 2014, 2017; He et al., 2017; Long et al., 2015), automatic segmentation methods still struggle with the segmentation of these intricate structures due to their complexity and low visibility. They often make topological errors such as broken connections or missing branches.

To address this, many turn to semi-automatic techniques, e.g., iterative proofreading by human annotators (Haehn et al., 2018), which can be time-consuming (Peng et al., 2011). This necessitates a better strategy to direct the annotators' attention towards locations that

© 2023 S. Gupta, X. Hu & C. Chen.

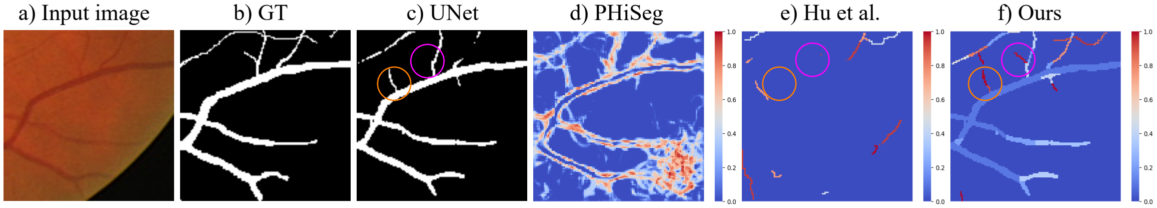

Figure 1: Motivating examples for structure-wise uncertainty. In the segmentation result (c), orange highlights a false positive structure, and pink highlights a false negative. Methods (d)-(f) are uncertainty estimates of the prediction in (c). PHiSeg (Baumgartner et al., 2019) assigns pixels along boundaries as uncertain. Hu et al. (Hu et al., 2023) captures uncertainty at a structural level, but produces overconfident maps (assigns zero uncertainty to many structures). Ours produces better structure-wise uncertainty estimates: both the highlighted false positive/negative structures have high uncertainty.

are more error-prone. By estimating the *uncertainty* (Gal et al., 2016), one can concentrate on the locations where a neural network is the least certain.

Despite many existing studies on segmentation uncertainty (Eaton-Rosen et al., 2018; Nair et al., 2020; Seeböck et al., 2019), most existing uncertainty estimation methods typically generate pixel-wise uncertainty maps which highlight pixels along the boundary of all structures as uncertain (see Fig. 1(d)). This offers limited information for human annotators; a desirable uncertainty map should instead highlight the error-prone "structures", e.g., small vessels/branches or short stretches of roads that tend to be disconnected or missed.

In this paper, we propose a new topology-aware uncertainty estimation method that highlights error-prone structures as a whole (such as in Fig. 1(f)). Highlighting structures with high uncertainty empowers annotators to accept or reject/correct structural proposals efficiently, thus streamlining the proofreading process. To capture the uncertainty of a given segmentation network's prediction at a structural level, we require the realization of two key components: a) decompose the prediction into a set of constituent structures, and b) estimate uncertainties of all the structures. We need to consider two types of structural uncertainty, intra-structural and inter-structural. The *intra-structural uncertainty* of a structure is due to its intrinsic composition, e.g., geometry, intensity, and the segmentation network's confidence. The *inter-structural uncertainty* is more contextual; it is due to interactions between neighboring structures. Our method explicitly models the two types of uncertainty.

We note that the method in Hu et al. (2023) (which we refer to as Hu et al.) also used DMT to decompose structures and estimate their uncertainty. However, their method employed coarse pruning, resulting in suboptimal uncertainty estimation. As illustrated in Fig. 1(e), Hu et al. produces overconfident maps; most structures, including many false negatives and false positives, are assigned zero uncertainty. In contrast, our method produces much better uncertainty estimates (Fig. 1(f)), owing to the proper modeling of both intra-structural and inter-structural uncertainties.

2. Related work

Topology-guided image segmentation. Several works focus on maintaining the correct connectivity or topology of thin structures. Topology-aware loss functions Mosinska et al.

(2018); Shit et al. (2021); Clough et al. (2020); Hu et al. (2019); Yang et al. (2021); Gupta et al. (2022); Hu (2022) impose per-pixel constraints to improve topological integrity. Discrete Morse theory has also been used to improve the topological awareness of segmentation networks Hu et al. (2021); Delgado-Friedrichs et al. (2014); Dey et al. (2019); Robins et al. (2011); Wang et al. (2015); Banerjee et al. (2020). These approaches use topological tools to improve segmentation at a pixel level, which is a weaker constraint compared to the structural level. In contrast, our method performs joint reasoning directly over the structures.

Uncertainty quantification. In recent years, there has been significant work on uncertainty quantification (UQ) of deep neural networks Abdar et al. (2021); Gawlikowski et al. (2021); Li et al. (2023). Here we review UQ techniques tailored for semantic segmentation. *Pixel-wise uncertainty:* Semantic segmentation is a per-pixel classification task and naturally most UQ methods produce per-pixel uncertainty estimates. In Kendall and Gal (2017), the authors propose a Bayesian framework using MC dropout Gal and Ghahramani (2015) and a learned loss attenuation to respectively capture model and data uncertainty. Recent methods have turned to generative models to generate multiple hypotheses, and the per-pixel variance across the hypotheses is treated as uncertainty. Some works in this direction are an ensemble of M networks Lakshminarayanan et al. (2017), a single network with M heads Rupprecht et al. (2017), Prob.-UNet Kohl et al. (2018), and PHiSeg Baumgartner et al. (2019). Prob.-UNet integrates a conditional variational autoencoder Sohn et al. (2015) with UNet Ronneberger et al. (2015), generating multiple hypotheses via latent variable sampling. PHiSeg extends this by introducing latent variables at every UNet level, thereby producing more diverse samples. *Structure-wise uncertainty:* Methods such as McClure et al. (2019); Seeböck et al. (2019) compute structure (volume) uncertainty by averaging over the pixel-wise uncertainty estimates. The method closest to ours is Hu et al. Hu et al. (2023). It is a generative model derived from Prob.-UNet where the latent variable has meaning in topology (specifically, a global persistence threshold). This threshold severely limits the structure space, overlooking several false positive/negative structures. Thus they tend to produce overconfident uncertainty estimates.

3. Method

Given a trained segmentation network, our goal is to capture the uncertainty of its prediction at a structural level. Note that we do not modify the network in any way; instead, we propose an external module that reasons the uncertainty of each structure in the segmentation.

Fig. 2 provides an overview of our method. Let F_θ denote the trained segmentation network, and M_ϕ denote our proposed external uncertainty quantification framework. M_ϕ takes as input the likelihood map of F_θ and the input image. It generates a set of structures, and estimates an uncertainty value for each of them. During training, M_ϕ is trained by comparing with the ground truth (GT) annotation.

M_ϕ consists of two primary modules to capture intra-structural and inter-structural uncertainty. The first module, Probabilistic DMT (Prob. DMT), generates structures based on the likelihood map. For each structure, it samples a set of skeletons representing different variations. The second module jointly predicts the uncertainties of all the structures. At each training iteration, it takes one sample skeleton for each structure, plus the likelihood map and input image, as input. More details of our method are described in Appendix A.

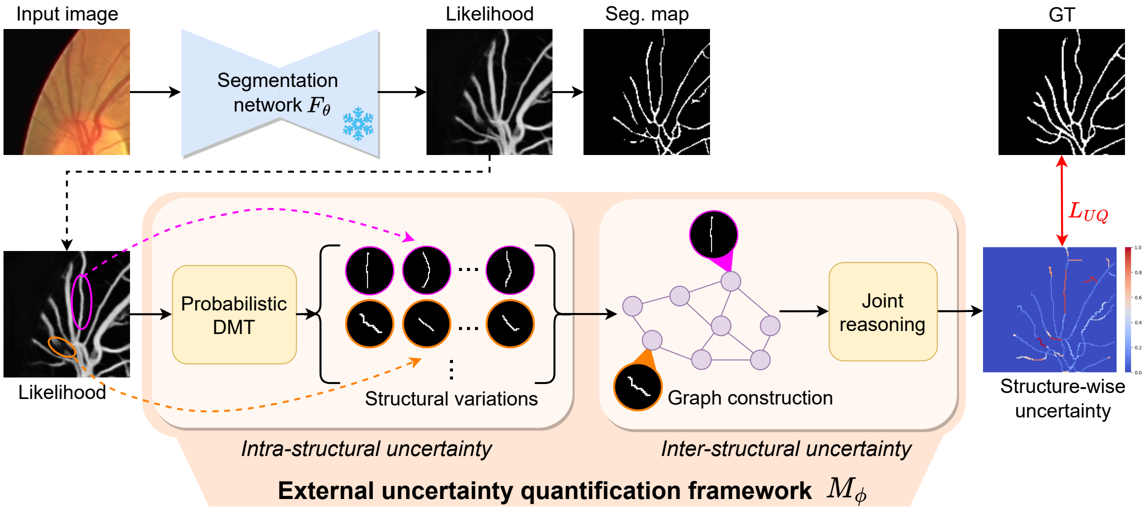
Input image
Segmentation network F_θ
Likelihood
Seg. map
GT
L_{UQ}
Likelihood
Probabilistic DMT
Structural variations
Graph construction
Joint reasoning
Structure-wise uncertainty
Intra-structural uncertainty
Inter-structural uncertainty
External uncertainty quantification framework M_ϕ

Figure 2: An overview of the proposed method M_ϕ. The given segmentation network F_θ has frozen weights. Probabilistic DMT decomposes the likelihood into structures, and samples skeleton representations of each. A graph is then constructed over the structures to perform joint reasoning of their uncertainty. The training is supervised by comparing with the GT.

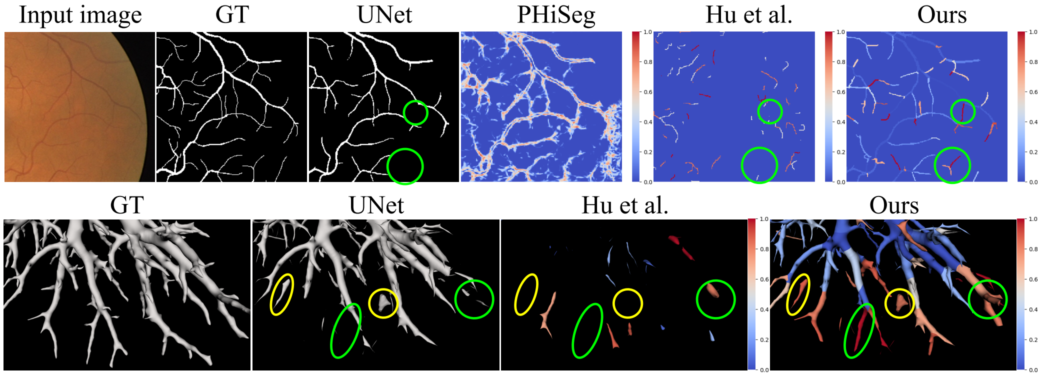

Figure 3: Qualitative results compared to the uncertainty baselines. We show uncertainty estimates in the form of a heatmap. Green highlights false negatives and yellow highlights false positives. Row 1: DRIVE; Row 2: PARSE (3D render).

4. Experiments

We broadly split our comparison baselines into two types: a) Pixel-wise uncertainty estimation methods: **Prob.-UNet** (Kohl et al., 2018), and **PHiSeg** (Baumgartner et al., 2019); b) Structure-wise uncertainty estimation method: **Hu et al.** (Hu et al., 2023). We evaluate our method on three datasets: **DRIVE** (Staal et al., 2004), **ROSE** (Ma et al., 2020), and **PARSE 2022 Grand Challenge** (Luo et al., 2023; Wang et al., 2022). To evaluate the uncertainty, we use **Expected Calibration Error (ECE)** (Naeini et al., 2015) and **Reliability Diagrams (RD)** (DeGroot and Fienberg, 1983). We also evaluate on segmentation and topological

Table 1: Comparison against uncertainty baselines on DRIVE (Staal et al., 2004) (all use UNet (Ronneberger et al., 2015) as the backbone)

Method	ECE (%)↓	Dice↑	clDice↑	ARI↑	VOI↓
Prob.-UNet	8.3316 ± 0.0043	0.7779 ± 0.0219	0.7663 ± 0.0492	0.7759 ± 0.0532	0.3560 ± 0.0203
PHiSeg	7.9316 ± 0.0032	0.7851 ± 0.0295	0.7712 ± 0.0497	0.7767 ± 0.0497	0.3527 ± 0.0308
Hu et al.	8.0883 ± 0.0036	0.7866 ± 0.0141	0.7725 ± 0.0392	0.7768 ± 0.0403	0.3489 ± 0.0286
Ours	**4.1633 ± 0.0043**	**0.7976 ± 0.0195**	**0.7974 ± 0.0372**	**0.7996 ± 0.0301**	**0.3322 ± 0.0229**

metrics such as **DICE** (Zou et al., 2004), **clDice** (Shit et al., 2021), **ARI** (Arganda-Carreras et al., 2015), and **VOI** (Meilă, 2007). More experimental details are in Appendix B.

5. Results

Tab. 1 shows the quantitative results against uncertainty methods, and Fig. 3 shows the respective qualitative results for the DRIVE dataset. Complete results can be found in Appendix C. Each table reports the mean and standard deviations for every metric, with statistically significant (Student, 1908) better performances in bold and numerically better (but not significant) performances in italics.

Performance of uncertainty estimation: Tab. 1 shows that our method outperforms others on both ECE and segmentation metrics. This is because we explicitly model the distribution of the structures, thereby quantifying the uncertainty of the segmentation network. In Fig. 3, we also see that our method generates better fidelity structure-wise uncertainty maps compared to Hu et al.

Performance of proofreading: One of the motivations of this work is to streamline the proofreading process. Structure-wise uncertainty can be used as a guide, with a user having to simply accept/reject a structure. We conduct experiments on the ROSE dataset and simulate user interaction with our method and Hu et al.'s. The user is given each method's final segmentation map, and inspects structures in decreasing order of uncertainty (till 0.5). Each uncertain structure is then subjected to a yes/no decision, which is denoted as one 'click'. The results are in Fig. 4. Our findings are consistent with the observation that Hu et al. assigns zero uncertainty to many structures; thus their margin of improvement is limited and saturates quickly.

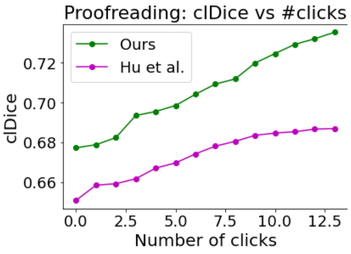

Figure 4: Proofreading.

6. Conclusion

In this abstract, we propose to quantify the structure-wise uncertainty of a given segmentation network. Our framework explicitly incorporates both intra-structural and inter-structural uncertainty, resulting in better fidelity uncertainty estimates. Our structure-wise uncertainty quantification can streamline the proofreading process by reducing the time spent finding and correcting errors. Extensive experiments show the practical applicability of our method over different segmentation backbones and datasets.

References

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 2021.

Ignacio Arganda-Carreras, Srinivas C Turaga, Daniel R Berger, Dan Cireşan, Alessandro Giusti, Luca M Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, 2015.

Samik Banerjee, Lucas Magee, Dingkang Wang, Xu Li, Bing-Xing Huo, Jaikishan Jayakumar, Katherine Matho, Meng-Kuan Lin, Keerthi Ram, Mohanasankar Sivaprakasam, et al. Semantic segmentation of microscopic neuroanatomical data by combining topological priors with encoder–decoder deep networks. *Nature machine intelligence*, 2020.

Christian F Baumgartner, Kerem C Tezcan, Krishna Chaitanya, Andreas M Hötker, Urs J Muehlematter, Khoschy Schawkat, Anton S Becker, Olivio Donati, and Ender Konukoglu. Phiseg: Capturing uncertainty in medical image segmentation. In *MICCAI*, 2019.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017.

James R Clough, Nicholas Byrne, Ilkay Oksuz, Veronika A Zimmer, Julia A Schnabel, and Andrew P King. A topological loss function for deep-learning based image segmentation using persistent homology. *TPAMI*, 2020.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 1983.

Olaf Delgado-Friedrichs, Vanessa Robins, and Adrian Sheppard. Skeletonization and partitioning of digital images using discrete morse theory. *TPAMI*, 2014.

Tamal K Dey, Jiayuan Wang, and Yusu Wang. Road network reconstruction from satellite images with machine learning supported by topological methods. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2019.

Zach Eaton-Rosen, Felix Bragman, Sotirios Bisdas, Sébastien Ourselin, and M Jorge Cardoso. Towards safe deep learning: accurately quantifying biomarker uncertainty in neural network predictions. In *MICCAI*, 2018.

Muhammad Moazam Fraz, Paolo Remagnino, Andreas Hoppe, Bunyarit Uyyanonvara, Alicja R Rudnicka, Christopher G Owen, and Sarah A Barman. Blood vessel segmentation methodologies in retinal images–a survey. *Computer methods and programs in biomedicine*, 2012.

Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

Yarin Gal et al. Uncertainty in deep learning. 2016.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021.

M González-Hidalgo. A survey on curvilinear object segmentation in multiple applications. *Pattern Recognition*, 2016.

Saumya Gupta, Xiaoling Hu, James Kaan, Michael Jin, Mutshipay Mpoy, Katherine Chung, Gagandeep Singh, Mary Saltz, Tahsin Kurc, Joel Saltz, et al. Learning topological interactions for multi-class medical image segmentation. In *ECCV*, 2022.

Daniel Haehn, Verena Kaynig, James Tompkin, Jeff W Lichtman, and Hanspeter Pfister. Guided proofreading of automatic segmentations for connectomics. In *CVPR*, 2018.

Tamir Hazan and Tommi Jaakkola. On the partition function and random maximum a-posteriori perturbations. In *ICML*, 2012.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

Xiaoling Hu. Structure-aware image segmentation with homotopy warping. *NeurIPS*, 2022.

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In *NeurIPS*, 2019.

Xiaoling Hu, Yusu Wang, Li Fuxin, Dimitris Samaras, and Chao Chen. Topology-aware segmentation using discrete morse theory. In *ICLR*, 2021.

Xiaoling Hu, Dimitris Samaras, and Chao Chen. Learning probabilistic topological representations using discrete morse theory. In *ICLR*, 2023.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R Ledsam, Klaus Maier-Hein, SM Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. In *NeurIPS*, 2018.

Rajitha Kv, Keerthana Prasad, and Prakash Peralam Yegneswaran. Segmentation and classification approaches of clinically relevant curvilinear structures: A review. *Journal of Medical Systems*, 2023.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Miguel Lazaro-Gredilla, Antoine Dedieu, and Dileep George. Perturb-and-max-product: Sampling and learning in discrete energy-based models. In *NeurIPS*, 2021.

Chen Li, Xiaoling Hu, and Chao Chen. Confidence estimation using unlabeled data. In *ICLR*, 2023.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

László Lovász. Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 1993.

Gongning Luo, Kuanquan Wang, Jun Liu, Shuo Li, Xinjie Liang, Xiangyu Li, Shaowei Gan, Wei Wang, Suyu Dong, Wenyi Wang, et al. Efficient automatic segmentation for multi-level pulmonary arteries: The parse challenge. *arXiv preprint arXiv:2304.03708*, 2023.

Yuhui Ma, Huaying Hao, Jianyang Xie, Huazhu Fu, Jiong Zhang, Jianlong Yang, Zhen Wang, Jiang Liu, Yalin Zheng, and Yitian Zhao. Rose: a retinal oct-angiography vessel segmentation dataset and new model. *TMI*, 2020.

Patrick McClure, Nao Rho, John A Lee, Jakub R Kaczmarzyk, Charles Y Zheng, Satrajit S Ghosh, Dylan M Nielson, Adam G Thomas, Peter Bandettini, and Francisco Pereira. Knowing what you know in brain segmentation using bayesian deep neural networks. *Frontiers in neuroinformatics*, 2019.

Marina Meilă. Comparing clusterings—an information based distance. *Journal of multivariate analysis*, 2007.

Volodymyr Mnih and Geoffrey E Hinton. Learning to detect roads in high-resolution aerial images. In *ECCV*, 2010.

Agata Mosinska, Pablo Marquez-Neila, Mateusz Koziński, and Pascal Fua. Beyond the pixel-wise loss for topology-aware delineation. In *CVPR*, 2018.

Lei Mou, Li Chen, Jun Cheng, Zaiwang Gu, Yitian Zhao, and Jiang Liu. Dense dilated network with probability regularized walk for vessel detection. *TMI*, 2019.

Lei Mou, Yitian Zhao, Huazhu Fu, Yonghuai Liu, Jun Cheng, Yalin Zheng, Pan Su, Jianlong Yang, Li Chen, Alejandro F Frangi, et al. Cs2-net: Deep learning segmentation of curvilinear structures in medical imaging. *MedIA*, 2021.

Kevin P Murphy. Conjugate bayesian analysis of the gaussian distribution. *def*, $(2\sigma 2)$, 2007.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, 2015.

Tanya Nair, Doina Precup, Douglas L Arnold, and Tal Arbel. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. *MedIA*, 2020.

George Papandreou and Alan L Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *ICCV*, 2011.

Hanchuan Peng, Fuhui Long, Ting Zhao, and Eugene Myers. Proof-editing is the bottleneck of 3d neuron reconstruction: the problem and solutions. *Neuroinformatics*, 2011.

William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 1971.

Vanessa Robins, Peter John Wood, and Adrian P Sheppard. Theory and algorithms for constructing discrete morse complexes from grayscale digital images. *TPAMI*, 2011.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *ICCV*, 2017.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 2008.

Philipp Seeböck, José Ignacio Orlando, Thomas Schlegl, Sebastian M Waldstein, Hrvoje Bogunović, Sophie Klimscha, Georg Langs, and Ursula Schmidt-Erfurth. Exploiting epistemic uncertainty of anatomy segmentation for anomaly detection in retinal oct. *TMI*, 2019.

Suprosanna Shit, Johannes C Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien PW Pluim, Ulrich Bauer, and Bjoern H Menze. cldice-a novel topology-preserving loss function for tubular structure segmentation. In *CVPR*, 2021.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. 2015.

Joes Staal, Michael D Abràmoff, Meindert Niemeijer, Max A Viergever, and Bram Van Ginneken. Ridge-based vessel segmentation in color images of the retina. *TMI*, 2004.

Student. The probable error of a mean. *Biometrika*, 1908.

Giles Tetteh, Velizar Efremov, Nils D Forkert, Matthias Schneider, Jan Kirschke, Bruno Weber, Claus Zimmer, Marie Piraud, and Bjoern H Menze. Deepvesselnet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes. *Frontiers in Neuroscience*, 2020.

Kuanquan Wang, Zhaowen Qiu, Wei Wang, Tao Song, Shaodong Cao, Yi Zhao, Jun Liu, Yingte He, Shaowei Gan, Xinjie Liang, Mingwang Xu, and Ziyu Guo. Pulmonary artery segmentation challenge 2022, March 2022. URL https://doi.org/10.5281/zenodo.6361906.

Suyi Wang, Yusu Wang, and Yanjie Li. Efficient map reconstruction and augmentation via topological methods. In *Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems*, 2015.

Jiaqi Yang, Xiaoling Hu, Chao Chen, and Chialing Tsai. A topological-attention convlstm network and its application to em images. In *MICCAI*, 2021.

Kelly H Zou, Simon K Warfield, Aditya Bharatha, Clare MC Tempany, Michael R Kaus, Steven J Haker, William M Wells III, Ferenc A Jolesz, and Ron Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports. *Academic radiology*, 2004.

— Appendix —

Appendix A provides more details about our method.

Appendix B provides experimental details such as dataset description, evaluation metrics, and implementation details.

Appendix C provides comprehensive qualitative and quantitative results of our method.

Appendix D provides details about the ablation study.

Appendix E discusses the broader impact and limitations of this work.

Appendix A. Method Details

Given a trained segmentation network, our goal is to capture the uncertainty of its prediction at a structural level. Note that we do not modify the network in any way; instead, we propose an external module that reasons the uncertainty of each structure in the segmentation.

Fig. 2 provides an overview of our method. Let F_θ denote the trained segmentation network, and M_ϕ denote our proposed external uncertainty quantification framework. M_ϕ takes as input the likelihood map of F_θ and the input image. It generates a set of structures, and estimates an uncertainty value for each of them. During training, M_ϕ is trained by comparing with the ground truth (GT) annotation.

M_ϕ consists of two primary modules to capture intra-structural and inter-structural uncertainty. The first module, Probabilistic DMT (Prob. DMT), generates structures based on the likelihood map. For each structure, it samples a set of skeletons representing different variations. Details are provided in Appendix A.1. The second module jointly predicts the uncertainties of all the structures. At each training iteration, it takes one sample skeleton for each structure, plus the likelihood map and input image, as input. Details are described in Appendix A.3. Throughout the sections, we consider one data sample (x, y) where x is an input image and y is the segmentation GT. The likelihood map is $f = F_\theta(x)$.

A.1. Modelling the structural space

In this section, we first describe how DMT obtains the constituent structures of a likelihood map. Then we propose our Prob. DMT formulation to capture intra-structural uncertainty.

Discrete Morse theory. Consider the likelihood map f generated from the segmentation network F_θ. We wish to decompose f into a set of structures, capturing not only the salient structures but also the faint ones. In the segmentation map, salient and faint structures broadly correspond to true positive and false negative structures. In Fig. 5(b), we highlight the false negative (FN) structures. These structures are missed in the segmentation, but will be captured by DMT (Fig. 5(d)).

DMT treats the likelihood map f as a terrain function, decomposing it into a *Morse complex* consisting of critical points, paths connecting them, patches in between paths, and volumes enclosed by patches (for 3D images). *Critical points* are locations w with zero gradients ($\nabla f(w) = 0$), i.e., minima, maxima, or saddle points.

Paths, called *V-paths*, are routes connecting critical points via the non-critical ones. A V-path connecting a saddle point to a maxima is called a stable manifold. These stable manifolds are the underlying terrain's mountain ridges, and delineate structures of interest.

In Fig. 5(c) we show the locations of saddles and maximas in the Morse complex, and in Fig. 5(d) we show the union of all the stable manifolds connecting them. In this paper, we only focus on the zero- and one-dimensional Morse structures, i.e., the union of all stable manifolds and their associated saddle and maxima. We call the collection of such structures the Morse skeleton.

By default, DMT generates stable manifolds in a completely deterministic manner, failing to take into account the intra-structural uncertainty in the likelihood f. Therefore, these stable manifolds may fail to correctly delineate the true structure, as shown in Fig. 6.

Probabilistic DMT.

To account for the inherent uncertainty, we explicitly model the structure as a collection of samples from an underlying generative process. The skeleton from the original DMT is just one possibility out of many. The method is achieved via a perturb-and-walk algorithm, in which we iteratively perturb the likelihood map, and regenerate the skeleton.

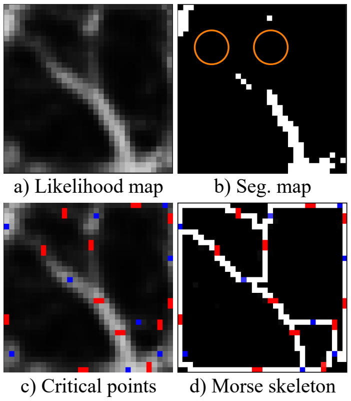

Figure 5: Orange indicates FN structures; (c) shows saddle points (red) and maximas (blue), and omits minimas; (d) shows the union of the stable manifolds of the saddle points.

The rationale is that the likelihood map is a weighted aggregation of all possible skeleton representations. To inverse the aggregation and recover these skeletons is challenging. Instead, we follow the classic *perturb-and-map principle*, which was used to efficiently sample from a complex discrete graphical model distribution Papandreou and Yuille (2011); Hazan and Jaakkola (2012); Lazaro-Gredilla et al. (2021). We randomly perturb the likelihood function. For each perturbed likelihood, we compute a skeleton as a sample. See Fig. 6 for an illustration.

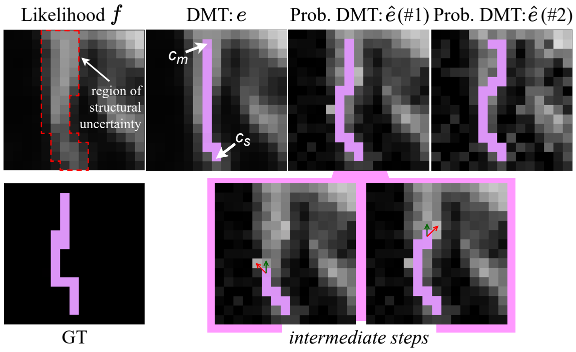
Likelihood f
DMT: e
Prob. DMT: $\hat{e}$ (#1)
Prob. DMT: $\hat{e}$ (#2)
region of structural uncertainty
c_m
c_s
GT
intermediate steps

Figure 6: Structures (#1,#2) sampled from the distribution. Green arrow is path chosen using $Q(c')$; red arrow is next step w/o considering $Q_d(c')$.

The sampled skeletons will reflect the uncertainty properly. For a structure that is less salient in the likelihood map, the sample skeletons will have large variations, generating a large uncertainty. For a salient structure in the likelihood map, the sample skeletons will be less variant, resulting in a low uncertainty.

Assume a given likelihood function f and one of its structures, represented by a V-path e connecting a saddle-maximum pair (c_s, c_m). We generate a sample skeleton of the structure by first perturbing the likelihood with random noise. Next, we generate a path connecting c_s and c_m. Recall in the original DMT, the skeleton is generated by following the mountain ridge. In other words, we start from the saddle point, and "walk" towards the maximum. At every step, we always walk to the neighboring pixel with the highest likelihood value. In Prob. DMT, we follow the same principle on the perturbed likelihood. However, the noisy perturbation of likelihood can cause the path to grow astray. Therefore, we additionally apply a distance-based regularizer to guide the walk towards the target c_m. We describe the process in detail below.

Let e denote the structure obtained by following the V-path between (c_s, c_m) in the original DMT. In order to generate its sample skeleton $\hat{e}$, we first draw a likelihood f_n from a distribution centered on f as $f_n \sim f + r$. This process is independent of the perturbation model r used, and we use a Gaussian model in this work. As the variance of the Gaussian model is unknown, we use Bayesian probability theory to sample the variance from the Inverse Gamma distribution (its conjugate prior (Murphy, 2007)).

Once we obtain f_n, we regenerate the path between (c_s, c_m). We take inspiration from random walk (Lovász, 1993) as well as probability regularized walk (Mou et al., 2019) to generate the variant structure $\hat{e}$ from f_n. Our walk algorithm continuously grows $\hat{e}$ starting from c_s and ending at c_m, one pixel at a time. The algorithm considers both the terrain f_n and the distance to the destination c_m to ensure path completeness. During the walk, given the current pixel location c, the next location $c' \in \text{neighborhood}(c)$[1] is chosen as $c' = \text{argmax}(Q(c'))$, where, $Q(c') = \gamma Q_d(c') + (1-\gamma) f_n(c')$, and, $Q_d(c') = \frac{1}{\|c_m - c'\|_2}$. We begin with $c \coloneqq c_s$ and continue in this manner till we reach c_m[2]. In Fig. 6, we show a deterministic structure obtained from DMT along with sample variations produced by our method. We demonstrate the intermediate steps in the algorithm: the red arrow denotes the next step without considering the distance regularizer Q_d, while the green arrow denotes the next step using our formulation Q. Notice how only considering f_n without Q_d can prevent the path from reaching c_m. We thus require Q_d to guide the path to completeness.

The structure $\hat{e}$ is a different realization of e, making each run of the Prob. DMT a stochastic one. We are thus able to explicitly model the structures as samples from a probability distribution. We also note that DMT is a special case of Prob. DMT when $r = 0$ and $\gamma = 0$. In practice, with some probability, we consider the original structure e from DMT over generating its variant $\hat{e}$. Specifically, following a Bernoulli distribution, with a small probability u we retain e, while with probability $1 - u$ we sample its variant $\hat{e}$ using the perturb-and-walk algorithm outlined above. This process is done separately and in parallel for every structure. The structures taken together form a Morse skeleton. The output of Prob. DMT is effectively one sample skeleton from the space of Morse skeletons.

A.2. Pseudocode of Probabilistic DMT

In Algo. 1, we provide a pseudocode of our Probabilistic DMT module proposed in Sec. A.1. We set *max_step* as 50 in our implementation. The terminologies used are: f^c denotes the likelihood map from F_θ centered on structure e; (c_s, c_m) are critical points between which the path e is generated: c_s denotes the saddle point and c_m denotes the maxima. IG denotes the Inverse Gamma distribution, and $\mathcal{N}$ denotes the Gaussian distribution.

[1] For neighborhood, we use 8-connectivity for 2D, and 26-connectivity for 3D in this work.

[2] If the path does not reach c_m, we impose a maximum limit on the update steps to prevent an infinite loop.

Algorithm 1 Probabilistic DMT pseudocode

```
procedure Prob_DMT($f^c, e, c_s, c_m$)
    $\hat{u} \sim$ Bernoulli($u$) if $\hat{u}$ is True then
        end
        $\hat{e} \leftarrow e$ else
        end
        $\hat{e} \leftarrow$ Generate_Path($f^c, c_s, c_m$)

    return $\hat{e}$
end procedure
procedure Generate_Path($f, c_s, c_m$)
    initialize $m \leftarrow 0$                    ▷ $m$ has same spatial dimension as $f$
    initialize $c \leftarrow c_s$
    initialize $m[c] \leftarrow 1$
    initialize $step \leftarrow 0$
    $\sigma^2 \sim IG(\alpha, \beta)$
    $f_n \sim f + \mathcal{N}(0, \sigma)$
        while $c \neq c_m$ and $step < max\_step$ do
        end
        $val \leftarrow 0$ for $c' \in Neighborhood(c)$ do
        end
        $val[c'] \leftarrow \gamma * \frac{1}{\|c_m - c'\|_2} + (1 - \gamma) * f_n[c']$

    $c \leftarrow$ argmax($val$)                   ▷ Update current step
    $m[c] \leftarrow 1$
    $step \leftarrow step + 1$

    return $m$
end procedure
```

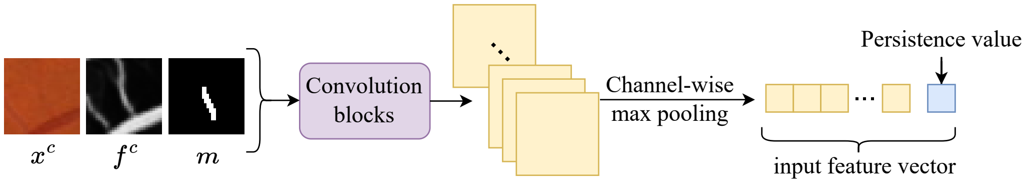

Figure 7: Construction of the input feature vector for each node (structure) in the GNN.

A.3. Joint estimation of structural uncertainty

The Prob. DMT module gives us a set E of structures. Our final step is to jointly reason about the uncertainty of all of them. To achieve this, we use a regression network that takes as input each structure $e \in E$, and outputs whether it is a true positive and the uncertainty of F_θ in predicting it.

Details of the network. Structures interact with each other in the image space and are not independent. During uncertainty estimation, it is therefore crucial to consider their spatial context, i.e., inter-structural uncertainty. Hence, we use Graph Neural Networks (Scarselli et al., 2008), specifically Graph Convolution Networks (GCN) (Kipf and Welling, 2016), to jointly reason about the structures and capture the high-order spatial interactions. In the graph, each node represents a structure, and edges between nodes exist when corresponding structures have non-zero overlaps (typically at endpoints). The input feature vector for each node is constructed as shown in Fig. 7. For every structure, we first concatenate $[x^c, f^c, m]$, where x^c comes from the original input x; f^c from the likelihood map f (not f_n); and m is a binary map indicating the presence of the structure. These x^c, f^c, m are smaller crops/bounding boxes centered on the structure. After passing them through convolution blocks, we apply channel-wise pooling to obtain a fixed-length feature vector for training. We further concatenate the persistence value of the saddle point associated with the original DMT structure (aka stable manifold). Note that we do not use the perturbed f_n from the Prob. DMT method when constructing the feature vector.

Training the network. We train the regression network using the attenuation loss proposed in (Kendall and Gal, 2017). As there are no labels to learn uncertainty, it is implicitly learned during regression optimization. We fix a Gaussian likelihood, and so variance $\hat{\delta}^2$ is used as a measure of uncertainty. The network's head is split into two — to predict $\hat{p}(e)$ of being a true positive structure and its associated uncertainty $\hat{\delta}^2_e$. For numerical stability, we actually predict the log variance $s_e = \log \hat{\delta}^2_e$. The training loss is given in Eq. 1. The structures that we obtain from Prob. DMT may not always fully overlap with the true GT structures, that is, some structures may only have partial overlap. We thus do not impose any hard constraints in Eq. 1, instead, z_e is a soft label, and is given by: $z_e = (\sum y \odot m)/(\sum m)$, where y is the GT and $\odot$ is the Hadamard product. This value simply represents the proportion of the structure that overlaps with the GT, i.e., the fraction of the structure that is a true positive.

$$L_{UQ}(\phi) = \frac{1}{|E|} \sum_{\forall e \in E} \left(\frac{1}{2} \frac{\|\hat{p}(e) - z_e\|^2}{\exp(s_e)} + \frac{1}{2} s_e \right) \qquad (1)$$

In (Kendall and Gal, 2017), $\hat{\delta}^2$ denoted the pixel-wise uncertainty of the framework's input. In our setting, the input to our framework is $f = F_\theta(x)$, and so $\hat{\delta}^2$ is modeled to capture the structure-wise uncertainty inherent in data x and model F_θ. Training $\hat{\delta}^2$ in this manner ensures that the network does not trivially predict high or low uncertainty, rather, predicts an uncertainty estimate that is dependent on the input.

A.4. Proposed module M_ϕ

For Eq. 1 to hold, we require M_ϕ to be a probabilistic network. We already show in Appendix A.1 our formulation for Prob. DMT. Additionally, the regression network is also probabilistic as we use MC dropout (Gal and Ghahramani, 2015).

Inference procedure. We take T runs of M_ϕ and compute the uncertainty as the mean $\bar{\delta}_e^2 = \frac{1}{T}\sum_{t=1}^{T}(\hat{\delta}_e^2)_t$. We similarly obtain $\bar{p}(e)$ from $\hat{p}(e)$. In Fig. 8, we illustrate the post-processing steps to obtain the structure-wise uncertainty heatmap. First, we obtain maps $\bar{p} = \cup \bar{p}_e$ and $\bar{\delta}^2 = \cup \bar{\delta}_e^2$ having the same spatial resolution as the input x. We then binarize $\bar{p}$, and overlay it onto the segmentation map obtained from F_θ. We do this because Prob. DMT gives us one-pixel wide skeleton structures but we need to recover the structure thickness. Next, we use shortest distance to assign uncertainty values from $\bar{\delta}^2$ to the pixels in the overlaid map. The shortest distance uses paths only along the foreground pixels.

In Fig. 8 we show how we obtain the final uncertainty heatmap from the skeleton heatmap. We also note that the overlaid map is an additional output of our method: it is an improved segmentation map that can be used instead of the one obtained by F_θ.

We illustrate the inference procedure in Fig. 9. There are two outcomes of our framework M_ϕ, namely, the structure-wise uncertainty heatmap as well as an improved discrete segmentation map that can be used instead of the one obtained by F_θ.

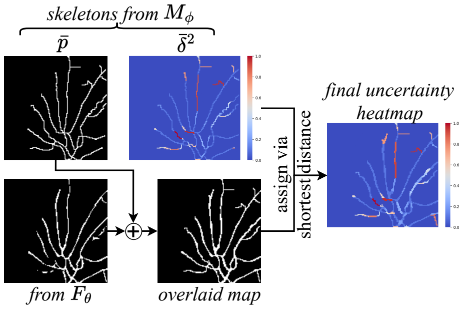

Figure 8: Post-processing procedure.

For each structure e, we obtain $\hat{p}(e)$ (the regression output) and $\hat{\delta}_e^2$ (the uncertainty) from M_ϕ. We take T runs of M_ϕ and then for each structure e, we compute the mean across T runs as $\bar{\delta}_e^2 = \frac{1}{T}\sum_{t=1}^{T}(\hat{\delta}_e^2)_t$, and, $\bar{p}(e) = \frac{1}{T}\sum_{t=1}^{T}\hat{p}(e)_t$. Next, we consider only those structures e for which $\bar{p}(e) \geq 0.5$, i.e., e has a minimum probability of 50% of being a true positive. This is the *threshold* step in Fig. 9. We do this so as to consider only the true positive, false positive, and false negative structures in the final outcomes. We use these structures to create a skeletal discrete segmentation map (see Fig. 9(c)) which has the same spatial resolution as Fig. 9(a). As we want to recover the thickness of each structure, we overlay the two maps to get the final discrete segmentation map (see Fig. 9(e)).

The uncertainty heatmap that we obtain from M_ϕ is also skeletal (see Fig. 9(d)). We recover the structure thickness to get the final uncertainty heatmap (see Fig. 9(f)). We use shortest distance to do this. Shortest distance is used to assign uncertainty values from Fig. 9(d) to the pixels in Fig. 9(e). The shortest distance uses paths only along the foreground pixels and not along the background ones. This ensures that pixels within a structure are not assigned uncertainty values from other nearby structures. We provide a zoomed-in view in Fig. 10.

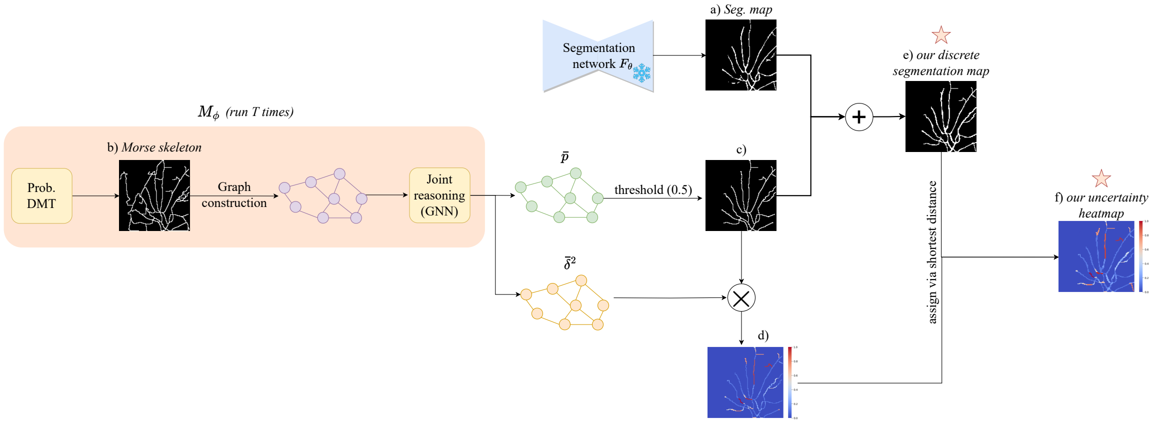

Figure 9: Inference procedure. Stars (⋆) denote the final outcomes of our framework M_ϕ.

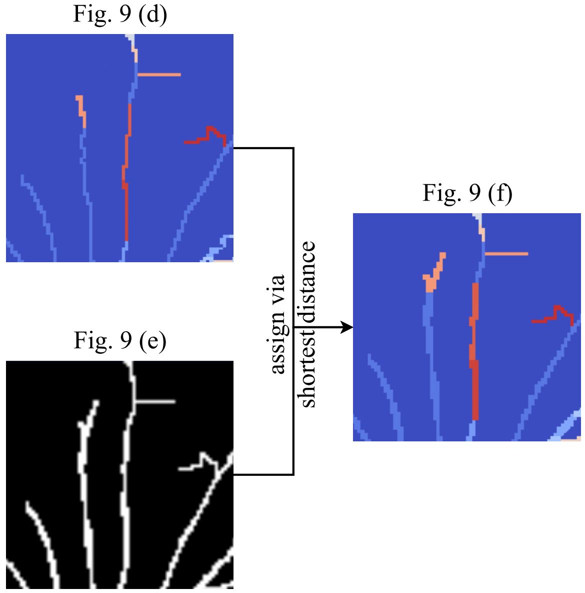

Figure 10: Zoomed-in views of Fig. 9.

We note that generating the Morse complex is computationally heavy, however, it needs to be computed only once across the T runs. As described in Appendix A.1, the sampled structures are between (c_s, c_m), and so the Morse complex is generated only in the first run.

A.5. Discussion of hyperparameters

The main hyperparameters in this work are u, α, β, γ. We describe the importance of each below:

- u: This is the parameter for the Bernoulli distribution. In our Prob. DMT module, for every structure, we have a choice to either retain the structure as obtained from DMT, or, generate a sample skeleton using the perturb-and-walk algorithm. We model this choice using the Bernoulli distribution. Essentially, in some runs we would like the original DMT structures to also interact with the others. Thus a low value of u works best. We found $u = 0.3$ to give the best performance, that is, for every structure there is a 30% chance that it's DMT form is used and a 70% chance that a sample variant is used. We find that $0.15 \leq u \leq 0.3$ have comparable performance.

- γ: This hyperparameter is used in the weighted combination of distance Q_d and likelihood f_n to obtain $Q(c')$, which is used to determine the next pixel location. It maintains a tradeoff between the distance regularizer Q_d and the perturbed likelihood f_n. The higher the value of γ, the greater the distance regularizer, and consequently the generated path will become closer to that of a straight line. This is not desirable, as a straight line would lose the original composition of the structure. Additionally, because of the perturbation in the likelihood, we do not want the path to go astray. To ensure path completeness, we require γ to be non-zero. Through experiments, we obtain the best performance when $\gamma = 0.2$.

- α, β: These are prior hyperparameters of the Inverse Gamma (IG) distribution. We perturb the likelihood using a Gaussian model. As the variance of the Gaussian model is unknown, we use Bayesian probability theory to sample the variance from the IG distribution (its conjugate prior). And so, α is the shape parameter and β is the scale parameter of this IG distribution. Ideally we would like a small perturbation of the likelihood and not a strong one. This is because a strong perturbation would corrupt wholly and we would not be able to sample a reasonable skeleton. At the same time, the perturbation should not be too small, otherwise we will not obtain a significant variant. The mean of the IG distribution is $\frac{\beta}{\alpha-1}$ (when $\alpha > 1, \beta > 0$), which on average is the value of the sampled variance for the Gaussian distribution. We achieve the best performance when $\alpha = 2.0$ and $\beta = 0.01$. The resulting sampled variance for the Gaussian model thus generates reasonable perturbation.

Appendix B. Experiment Details

B.1. Dataset Details

In this abstract, we validate our results on three segmentation datasets: DRIVE (Staal et al., 2004), ROSE (Ma et al., 2020), and PARSE 2022 Grand Challenge (Wang et al., 2022).

DRIVE. The DRIVE dataset is a 2D retinal vessel dataset with 40 images. Each image has a resolution of 584×565. We use the dataset's predetermined split of 20 training images and 20 test images. For training, we keep aside four randomly-chosen samples as validation, and train on the remaining 16 samples.

ROSE. The ROSE dataset is a 2D retinal OCTA (Optical Coherence Tomography Angiography) segmentation dataset. We use ROSE-1 (SVC) in this work. It has a predetermined split of 30 train and 9 test samples, with each sample having a resolution of 304×304. For training, we keep aside four randomly-chosen samples as validation, and train on the remaining 26 samples.

PARSE. The PARSE dataset is a 3D CT dataset containing pulmonary artery segmentations. The dataset contains 100 volumes and their sizes vary from $512 \times 512 \times 228$ to $512 \times 512 \times 376$.

As there is no predetermined train/test split, we use 4-fold cross-validation and report the average performance.

B.2. Evaluation Metrics

We use both segmentation and uncertainty metrics to evaluate our method. We describe the metrics in detail below.

B.2.1. Uncertainty metrics

We use Reliability diagrams (DeGroot and Fienberg, 1983) and Expected calibration error (ECE) (Naeini et al., 2015) to evaluate the quality of uncertainty. As both the metrics were originally designed for classification, we adapt from the classification task to semantic segmentation by treating each pixel as an independent sample. For both metrics, we first divide the probability interval $[0, 1]$ into N equal-sized probability intervals (each of size $\frac{1}{N}$). We use $N = 20$ bins in this work. We then calculate the *accuracy* and *confidence* of each bin.
Reliability diagrams (RD). Reliability diagrams (DeGroot and Fienberg, 1983) are a visual representation of model calibration by plotting the expected accuracy as a function of confidence ($confidence = 1 - uncertainty$). Perfect calibration corresponds to an identity function in the RD, i.e., the model is not over/under-confident. Consider the set of pixels/structures whose predicted probabilities fall into the bin B_i. The accuracy and confidence are given by:

$$acc(B_i) = \frac{1}{|B_i|} \sum_{\forall x \in B_i} \mathbf{1}\left(\hat{Y}(x) = Y(x)\right)$$

$$conf(B_i) = \frac{1}{|B_i|} \sum_{\forall x \in B_i} \hat{P}(x)$$

where, Y is the discrete segmentation ground truth (GT), and $\hat{Y}$ is the discrete segmentation map outputted by the model. In our method, $\hat{Y}$ is as shown in Fig. 9(c). Additionally, $\hat{P}$ is the pixel-wise probability (likelihood) outputted by the model, whereas in our case, it is the structure-wise uncertainty $\bar{\delta}^2$ (Fig.9(d)). For our method and Hu et al., the $x \in B_i$ denotes structures, while in the other methods, it denotes pixels.
Expected calibration error (ECE). RDs are only a visual cue, and so we also use ECE (Naeini et al., 2015): a scalar to summarize the calibration performance. RDs do not take into account the number of pixels/structures in each bin. Thus, to account for such variations of the number of samples in a bin, we use ECE. It is given by:

$$ECE = \sum_{i=1}^{N} \frac{|B_i|}{n} |acc(B_i) - conf(B_i)|$$

where $n = \sum_i^N |B_i|$ is the total number of pixels/structures. The difference between acc and $conf$ for a given bin represents the calibration gap. When there is perfect calibration, ECE is zero.

The definition of acc and $conf$ remains the same as defined for RDs.

Table 2: Training configuration

Dataset	Model	Patch Size; Batch Size	Learning rate (LR); Optimizer
DRIVE	UNet	$128 \times 128; 8$	1e-3 with LR scheduler[3]; Adam[4] with weight decay 3e-5
	DeepVesselNet	$128 \times 128; 8$	1e-3 with LR scheduler; Adam with weight decay 3e-5
	CS^2-Net	$128 \times 128; 8$	1e-3 with LR scheduler; Adam with weight decay 3e-5
	Prob.-UNet	$128 \times 128; 8$	1e-3; Adam with weight decay 0
	PhiSeg	$128 \times 128; 8$	1e-4 with LR scheduler; Adam with weight decay 1e-5
	Hu et al.	$128 \times 128; 8$	1e-3; Adam with weight decay 0
	Ours	$128 \times 128; 8$	1e-3; Adam with weight decay 0
ROSE	UNet	$128 \times 128; 6$	1e-3 with LR scheduler; Adam with weight decay 3e-5
	DeepVesselNet	$128 \times 128; 6$	1e-3 with LR scheduler; Adam with weight decay 3e-5
	CS^2-Net	$128 \times 128; 6$	1e-3 with LR scheduler; Adam with weight decay 3e-5
	Prob.-UNet	$128 \times 128; 6$	1e-3; Adam with weight decay 0
	PhiSeg	$128 \times 128; 6$	1e-4 with LR scheduler; Adam with weight decay 1e-5
	Hu et al.	$128 \times 128; 6$	1e-3; Adam with weight decay 0
	Ours	$128 \times 128; 6$	1e-3; Adam with weight decay 0
PARSE	UNet	$128 \times 128; 8$	1e-3 with LR scheduler; Adam with weight decay 3e-5
	DeepVesselNet	$128 \times 128; 8$	1e-3 with LR scheduler; Adam with weight decay 3e-5
	CS^2-Net	$128 \times 128; 8$	1e-3 with LR scheduler; Adam with weight decay 3e-5
	Prob.-UNet	$128 \times 128; 8$	1e-3; Adam with weight decay 0
	PhiSeg	$128 \times 128; 8$	1e-4 with LR scheduler; Adam with weight decay 1e-5
	Hu et al.	$128 \times 128; 8$	1e-3; Adam with weight decay 0
	Ours	$128 \times 128; 8$	1e-3; Adam with weight decay 0

B.2.2. Segmentation metrics

DICE. DICE (Zou et al., 2004) score is a popular metric which measures the area/volumetric overlap between the predicted and ground truth discrete masks. It overcomes the class imbalance problem in the pixel-wise accuracy metric by considering only the foreground classes for measuring the overlap. The higher the DICE, the better the segmentation.
clDice. clDice (Shit et al., 2021) is derived from DICE, however, clDice uses the skeleton of the predictions. This makes it sensitive to the performance of thin structures like vessels which is important in curvilinear segmentation. The higher the value, the better the segmentation.
ARI. The Rand index (Rand, 1971) computes similarity between two clustering. This raw score is “adjusted for chance" to get ARI (Adjusted Rand Index) (Arganda-Carreras et al., 2015). The ARI takes into account the fact that some agreement between the two clusterings can occur by chance, and it adjusts the Rand index to account for this possibility. The higher the value, the better the segmentation.
VOI. The VOI (Meilă, 2007) metric is defined as the sum of the conditional entropies between two segmentations. A lower VOI value indicates better segmentation.

B.3. Implementation Details

We use the PyTorch framework, a single NVIDIA Tesla V100-SXM2 GPU (32G Memory) and a Dual Intel Xeon Silver 4216 CPU@2.1Ghz (16 cores) for all the experiments. The training hyperparameters for our method as well as the baselines are as tabulated in Tab. 2. Note that although PARSE is a 3D dataset, all the segmentation networks (backbones) F_θ are 2D, that is, the networks are trained on 2D slices of the dataset. This was done to

[3] https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

[4] https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

maintain a fair comparison across all baselines, as methods such as Prob.-UNet and PhiSeg only had 2D implementations available.

B.3.1. Baselines

We use the publicly available codes for the baselines:

- UNet (Ronneberger et al., 2015): https://github.com/johschmidt42/PyTorch-2D-3D-UNet-Tutorial
- Prob.-UNet (Kohl et al., 2018): https://github.com/stefanknegt/Probabilistic-Unet-Pytorch
- PhiSeg (Baumgartner et al., 2019): https://github.com/gigantenbein/UNet-Zoo
- Hu et al. (Hu et al., 2023): https://github.com/HuXiaoling/Structural_Uncertainty
- DeepVesselNet (Tetteh et al., 2020): https://github.com/dhavalshah18/deepvesselnet
- CS^2-Net (Mou et al., 2021): https://github.com/iMED-Lab/CS-Net

B.3.2. Our method

We plan to release our code upon acceptance. For reproducibility, we provide the architecture details as follows. The 'Joint reasoning' module in our framework is a Graph Neural Network (GNN) (Scarselli et al., 2008), specifically Graph Convolution Network (GCN) (Kipf and Welling, 2016). As per Fig. 7, the input feature vector for each graph node is constructed by passing $[x^c, f^c, m]$ through the following architecture: $C(3,24) \rightarrow ReLU \rightarrow D(0.2) \rightarrow C(3,32) \rightarrow ReLU \rightarrow D(0.2) \rightarrow MaxPool \rightarrow Concat(pers)$, where, $C(a,b)$ denotes a convolution layer having kernel size a and number of output channels b; $D(p)$ denotes a Dropout layer with probability p; $MaxPool$ denotes the adaptive maxpool layer[5] returning a 1×1 output for each channel; and $pers$ is a scalar value denoting the persistence of the structure. Furthermore, the bounding box size of $[x^c, f^c, m]$ is 32×32 centered at each structure.

The aforementioned layers generate the input feature vector for each graph node. They are then passed through the GNN which contains the following layers: $GCN(32) \rightarrow ReLU \rightarrow D(0.2) \rightarrow GCN(64) \rightarrow ReLU \rightarrow D(0.2) \rightarrow GCN(32) \rightarrow ReLU \rightarrow D(0.2)$, where $GCN(a)$ denotes a GCNConv layer [6] having a number of output channels. The output from this sequence of layers is then fed to two separate $GCN(1)$ layers to output the regression likelihood $\hat{p}(e)$ and the uncertainty $\hat{\delta}_e^2$. As per GNN fashion, the weights of the layers are shared across all the nodes.

Appendix C. Results

Tab. 3 shows the quantitative results against uncertainty methods, and Tab. 4 shows the quantitative results on different backbone architectures. We show the respective qualitative results in Fig. 11 and Fig. 12. We also perform the unpaired **t-test** (Student, 1908) (95%

[5] https://pytorch.org/docs/stable/generated/torch.nn.AdaptiveMaxPool2d.html

[6] https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GCNConv.html#torch_geometric.nn.conv.GCNConv

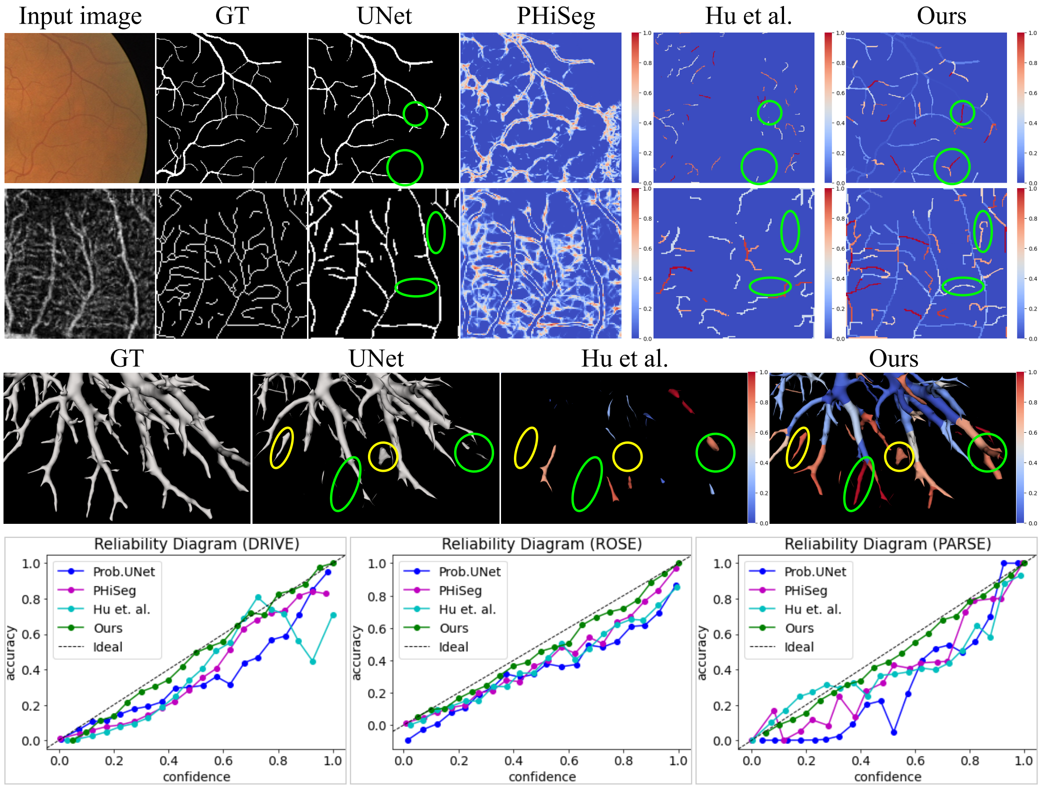

Figure 11: Qualitative results compared to the uncertainty baselines. We show reliability diagrams of the samples, and uncertainty estimates in the form of a heatmap. Green highlights false negatives and yellow highlights false positives. Row 1: DRIVE; Row 2: ROSE; Row 3: PARSE (3D render).

confidence interval) to determine the statistical significance. Each table reports the mean and standard deviations for every metric, with statistically significant better performances in bold and numerically better (but not significant) performances in italics. For all the probabilistic methods, the average of five runs was used. For our method, we generated the structure-wise uncertainty estimates and the segmentation map by following the steps outlined in the 'Inference procedure' in Appendix A.4. We discuss the performances below.

Performance of uncertainty estimation. Tab. 3 shows that our method outperforms others on both ECE and segmentation metrics. Fig. 11 displays RDs, with our method following the ideal line much closely compared to others. This is because we explicitly model the distribution of the structures, thereby quantifying the uncertainty of the segmentation network. In Fig. 11, we also see that our method generates better fidelity structure-wise uncertainty maps compared to Hu et al. Our heatmaps assign non-zero uncertainty to several false positives/negatives in the backbone UNet's outputs. This is because we reason about every structure while Hu et al. limits the structure space via pruning.

Performance over different backbones. Tab. 4 and Fig. 12 show that our method is backbone-agnostic. It consistently improves the segmentation quality and produces high fidelity uncertainty maps for each of the underlying networks. This validates the practical applicability of our method. Results using the UNet backbone can be found in Tab. 3.

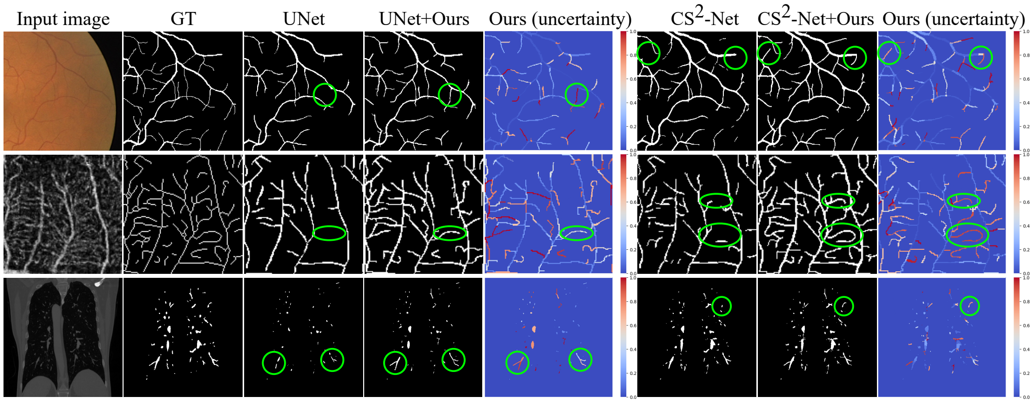

Figure 12: Qualitative results over different segmentation backbones. Green highlights false negatives. Row 1: DRIVE; Row 2: ROSE; Row 3: PARSE.

Table 3: Comparison against uncertainty baselines (all use UNet (Ronneberger et al., 2015) as the backbone)

Dataset	Method	ECE (%)↓	Dice↑	clDice↑	ARI↑	VOI↓
DRIVE	Prob.-UNet	8.3316 ± 0.0043	0.7779 ± 0.0219	0.7663 ± 0.0492	0.7759 ± 0.0532	0.3560 ± 0.0203
	PHiSeg	7.9316 ± 0.0032	0.7851 ± 0.0295	0.7712 ± 0.0497	0.7767 ± 0.0497	0.3527 ± 0.0308
	Hu et al.	8.0883 ± 0.0036	0.7866 ± 0.0141	0.7725 ± 0.0392	0.7768 ± 0.0403	0.3489 ± 0.0286
	Ours	**4.1633 ± 0.0043**	**0.7976 ± 0.0195**	**0.7974 ± 0.0372**	**0.7996 ± 0.0301**	**0.3322 ± 0.0229**
ROSE	Prob.-UNet	7.2795 ± 0.0022	0.7378 ± 0.0284	0.6485 ± 0.0258	0.7219 ± 0.0538	0.7769 ± 0.0146
	PHiSeg	7.0875 ± 0.0036	0.7415 ± 0.0267	0.6552 ± 0.0236	0.7309 ± 0.0425	0.7638 ± 0.0128
	Hu et al.	6.9243 ± 0.0033	0.7429 ± 0.0132	0.6598 ± 0.0172	0.7506 ± 0.0302	0.7616 ± 0.0123
	Ours	**3.9904 ± 0.0041**	**0.7593 ± 0.0171**	**0.6782 ± 0.0119**	**0.7837 ± 0.0314**	**0.7403 ± 0.0239**
PARSE	Prob.-UNet	9.9918 ± 0.0069	0.6002 ± 5.7751	0.6179 ± 0.0804	0.6523 ± 0.0654	0.8923 ± 0.0417
	PHiSeg	9.9280 ± 0.0077	0.5910 ± 3.0858	0.6080 ± 0.0743	0.6512 ± 0.0521	0.8839 ± 0.0297
	Hu et al.	7.7891 ± 0.0075	0.6044 ± 2.3583	0.6153 ± 0.0724	0.6537 ± 0.0363	0.8803 ± 0.0318
	Ours	**4.0289 ± 0.0073**	*0.6190 ± 3.0826*	**0.6221 ± 0.0613**	**0.6658 ± 0.0461**	**0.8701 ± 0.0332**

Table 4: Comparison against different segmentation backbones

Dataset	Method	Dice↑	clDice↑	ARI↑	VOI↓
DRIVE	DeepVesselNet	0.8015 ± 0.0260	0.7997 ± 0.0431	0.7729 ± 0.0457	0.3413 ± 0.0256
	DeepVesselNet + Ours	**0.8173 ± 0.0190**	**0.8285 ± 0.0361**	**0.8037 ± 0.0361**	**0.3238 ± 0.0192**
	CS^2-Net	0.8189 ± 0.0176	0.8125 ± 0.0413	0.8204 ± 0.0495	0.3417 ± 0.0203
	CS^2-Net + Ours	**0.8301 ± 0.0172**	**0.8367 ± 0.0305**	**0.8495 ± 0.0301**	**0.3243 ± 0.0258**
ROSE	DeepVesselNet	0.7653 ± 0.0101	0.6634 ± 0.0192	0.7622 ± 0.0302	0.7426 ± 0.0163
	DeepVesselNet + Ours	**0.7795 ± 0.0205**	**0.6873 ± 0.0195**	**0.7936 ± 0.0282**	**0.7164 ± 0.0226**
	CS^2-Net	0.7623 ± 0.0285	0.6799 ± 0.0127	0.7702 ± 0.0322	0.7236 ± 0.0157
	CS^2-Net + Ours	**0.7886 ± 0.0208**	**0.6968 ± 0.0149**	**0.7981 ± 0.0211**	**0.7072 ± 0.0168**
PARSE	DeepVesselNet	0.7208 ± 3.0452	0.6801 ± 0.0554	0.6923 ± 0.0524	0.4907 ± 0.0701
	DeepVesselNet + Ours	*0.7376 ± 3.1863*	**0.6983 ± 0.0622**	**0.7098 ± 0.0613**	**0.4711 ± 0.0613**
	CS^2-Net	0.7630 ± 3.9415	0.6918 ± 0.0695	0.7138 ± 0.0695	0.4273 ± 0.0521
	CS^2-Net + Ours	*0.7720 ± 2.8109*	**0.7113 ± 0.0689**	**0.7343 ± 0.0733**	**0.4078 ± 0.0642**

Appendix D. Ablation Study

To demonstrate the efficacy of the proposed method, we conduct ablation studies of the different components in our pipeline, as well as check the effect of changing hyperparameter values. We perform these analyses on the DRIVE dataset using UNet (Ronneberger et al., 2015) as the backbone.

Table 5: Ablation of different modules

DMT	Reg. Net	ECE (%)↓	clDice↑
DMT	GNN	6.3481 ± 0.0082	0.7729 ± 0.0304
Prob. DMT	MLP	4.8202 ± 0.0046	0.7745 ± 0.0305
Prob. DMT	GNN	**4.1633 ± 0.0043**	**0.7974 ± 0.0372**

Ablation of different modules. We conduct ablation studies on both parts of our framework: structure generation (DMT vs Prob. DMT), and regression network (GNN vs Multi-layer Perceptron (MLP)). The results are in Tab. 5.

Prob. DMT results in a sharp improvement in ECE compared to the original DMT; this supports our assertion that Prob. DMT models intra-structural uncertainty. Similarly, using GNN over MLP results in improvement. The message-passing in GNNs accounts for inter-structural uncertainty, thus yielding higher fidelity uncertainty estimates.

Effect of hyperparameters. We check the effect of the different hyperparameters in our work by conducting experiments on the DRIVE dataset using UNet as the backbone. Our main hyperparameters are u, α, β, γ, with u used in the Bernoulli distribution, γ in the path-generation algorithm, and (α, β)[7] as prior hyperparameters of the Inverse Gamma distribution. We test different values and report the ECE (the lower the better) in Fig. 13. For all the experiments, we set $u = 0.3$. We achieve the best ECE when $\gamma = 0.2, \alpha = 2.0, \beta = 0.01$, however, a reasonable range always yields improvement (notice how non-zero γ results in a sharp improvement). This demonstrates the robustness of our proposed method.

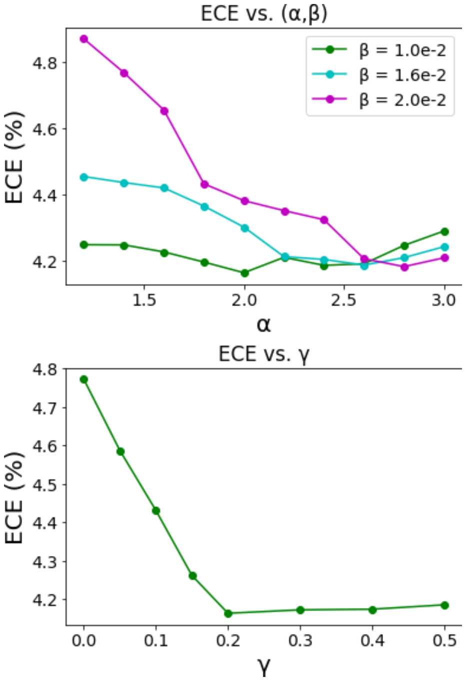

Figure 13: Effect of hyperparameters.

Appendix E. Broader impact and Limitations

Broader impact In this work, we aim to capture structure-wise uncertainty of a given network, where a structure is defined to be a coherent set of pixels a user can intuitively understand, e.g., small vessels/branches, short stretches of road etc. Fine-scale structures such as vessels, neurons, and membranes often consist of interconnected branches or structures that form a cohesive entity. Thus structure-wise uncertainty maps can highlight uncertain instances or branches as a whole, providing a more accurate indication of regions where the segmentation may be inaccurate or uncertain. This is beneficial for proofreading or error-correction tasks as they can direct the focus of human annotators to uncertain structures that

[7] α is the shape parameter; β is the scale parameter.

require further attention. This can save time and effort compared to pixel-wise uncertainty maps that highlight numerous pixels as uncertain, many of which do not require correction. Thus structure-wise uncertainty can provide more interpretable estimates and is a desirable approach for improving segmentation accuracy and supporting downstream analysis tasks. This can go a long way as the benefit of proofreading is twofold: it improves segmentation quality, and it also helps expand the body of labeled data that can be further used to train automatic segmentation methods. Our work is thus a useful tool in streamlining the process of scalable annotation. At the present stage, we do not foresee any potential negative societal impacts.

Limitations Our method currently fits in the context of curvilinear segmentation. In general, large object segmentation could also benefit from structure-wise uncertainty (structures in this case would be smaller patches/volumes). Discrete Morse theory can be used in this setting, however, we would need to make use of topological features other than the stable manifold. In its present form, our proposed solution is currently not applicable in a setting beyond curvilinear segmentation.