Physical Constraints-Aware Machine Learning Models for Vascular Image Analysis

Suprosanna Shit

TUM School of Computation, Information and Technology
Technische Universität München

Physical Constraints-Aware Machine Learning Models for Vascular Image Analysis

Suprosanna Shit

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:
Prof. Dr. Nils Thürey

Prüfer*inen der Dissertation:
1. Prof. Dr. Bjoern Menze
2. Prof. Nils Daniel Forkert, Ph.D.
3. Prof. Dr. Bjoern Andres

Die Dissertation wurde am 11.01.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 27.07.2023 angenommen.

Dedicated to my parents, teachers and friends.

Abstract

Evolution has preferred a network strategy, namely vascular systems, for the internal logistics of living beings. Any disruptions in the expected behavior of the vascular system impact the functional normality from cells to organs to the individual level, even sometimes fatal. Hence, the study of the vascular system is of immense urgency in medical and biological science. The main research direction embraces identifying structural and functional properties of vascular systems, their inter-dependencies, and the appearance of pathogens in the system. This scientific curiosity is met with modern tools developed with the help of other branches of science, such as physics and engineering, which enable photographing the vascular system in action. For example, the functional imaging modalities considered in this thesis are phase-contrast magnetic resonance imaging to measure blood flow velocity within vessels and light sheet microscopy for structural imaging, which allows imaging up to the capillary level. A natural next step is to analyze these images, which modern deep learning is getting very good at. The aim of this thesis is to advance deep learning methodologies and sharpen them with the necessary features to deal with challenges in vascular imaging. Specifically, this thesis considers the different genres of constraints that occur from the underlying nature of the biophysical systems and adopts them in deep learning settings. In these crossroads, the core of this dissertation consists of four contributions. 1) The first project marries implicit neural functions and convolutional neural networks to efficiently enforce partial differential equations for obtaining the pressure distribution within blood vessels from measured blood flow velocity. 2) Next, a directional sensitive loss function is introduced to emphasize the directional correctness, which is suitable for super-resolving the flow velocities. 3) Third, an efficient loss function is proposed to highlight the important pixels in an image, which are crucial to preserving the underlying network topology in the segmented vasculature. 4) Finally, a novel model architecture is designed to directly learn the underlying graph representation vessel-network from voxel representation of vascular images. In summary, this thesis contributes to the theoretical and experimental advancement of the functional and structural analysis of blood vessels from contemporary vascular imaging modalities.

Zusammenfassung

Während der Evolution hat sich eine netzwerkartige Gefäßstruktur für die interne Logistik in Lebewesen entwickelt und durchgesetzt. Jede Störung des sensiblen Gefäßsystems wirkt sich auf die Funktionalität von Zellen, Organen bis hin zu ganzen Individuen aus. Daher ist die Untersuchung des Gefäßsystems in der medizinischen und biologischen Wissenschaft von immenser Relevanz. Insbesondere ist die Identifizierung der strukturellen und funktionellen Eigenschaften von Gefäßsystemen und die Beschreibung ihrer gegenseitigen Abhängigkeiten interessant. Das wissenschaftliche Interesse an Gefäßen wird durch die Entwicklung moderner Bildgebung Instrumente und Methoden unter Mithilfe anderer Wissenschaftszweige, wie der Physik und der Ingenieurwissenschaften begünstigt. Die in dieser Arbeit betrachteten funktionellen Bildgebungsmodalitäten sind die Phasenkontrast-Magnetresonanztomographie zur Messung der Blutflussgeschwindigkeit in den Gefäßen, und die Lightsheet Mikroskopie für die strukturelle Bildgebung, welche eine Bildgebung bis auf Kapillarebene ermöglicht. Ein natürlicher nächster Schritt ist die automatisierte Analyse dieser Bilder durch modernes Deep Learning. Ziel dieser Arbeit ist es, Deep-Learning-Methoden weiterzuentwickeln und sie mit dem nötigen Rüstzeug auszustatten, um die Herausforderungen der modernen vaskulären Bildgebung zu bewältigen. Insbesondere werden in dieser Arbeit biophysikalische Bedingungen und Eigenschaften berücksichtigt und für unsere lernenden Methoden nutzbar gemacht. Der Kern dieser Dissertation besteht aus vier Beiträgen. 1) Im ersten Projekt werden partielle Differentialgleichungen anhand von impliziten neuronalen Funktionen und Neuronale Netzen gelößt, um die Druckverteilung in Blutgefäßen anhand gemessener Blutflussgeschwindigkeit zu ermitteln. 2) Als nächstes wird eine richtungssensitive Optimierungsfunktion eingeführt die für die Superauflösung der Flussgeschwindigkeiten von Blut geeignet ist. 3) Drittens wird eine effiziente Optimierungsfunktion für Segmentierung von Blutgefäß vorgeschlagen, welche die wichtigen Pixel in einem Bild hervorhebt die für die Erhaltung der Netzwerktopologie von gefäßen entscheidend sind. 4) Abschließend wird eine neuartige Netzwerkarchitektur entworfen, um aus Bildern eine Graphendarstellung von Gefäßnetzwerken zu erlangen. Zusammenfassend leistet diese Arbeit einen Beitrag zur theoretischen und experimentellen Weiterentwicklung der funktionellen und strukturellen Analyse von Blutgefäßen mit modernen bildgebenden Verfahren.

Acknowledgements

Pursuing doctoral studies has been a remarkable professional journey and an unprecedented opportunity for personal growth, especially during the pandemic. And when it comes to writing about people who have influenced, supported, tolerated, and celebrated me, it is hard to keep emotions aside and express them in words. Nevertheless, I will make an attempt here. The one who enabled this journey in the first place and held my hand strong when I stumbled is none other than my supervisor. Thank you, Bjoern, for everything you have provided; for all the freedom of thought, academic training, brainstorming sessions, countless back and forth of manuscripts, all those lunchtime chat, everything. I feel blessed to have you as a boss, and I couldn't have asked more.

My next support system comes from my incredible peers. We spent so much time together writing so many papers, working on numerous projects, helping each other, sharing sleepless nights to meet the deadline, with the same energy level, hiking together at so many peaks, traveling so many different places, chilling so many afternoons beside the river, infinite philosophical discussion, so many memories that it is really difficult to single out any particular occasion. Johannes, Ivan, Anjany, Fercho, and Bran, thank you, mates, for being into this together. Suffering, rejoicing and experiencing life at its finest. I feel tremendously fortunate to have highly talented colleagues turned friends around me, who have shaped my academic path in many ways. Thank you, Dhritiman, Judith, Florian, Oliver, Giles, Amir, Jana, Carolin, Diana, John, Sebastian, Rami, Izabela, Chinmay, and Bastian. Special mention to Marie and Augusto for helping me navigate the early days of my Ph.D.

The last one and a half years have been unexpectedly special, being adopted into Daniel's group. Thank you, Daniel, for all the support and warm-hearted welcome to your academic family. Alina, Jiazhen, Özgün, Leo, Martin, Dima, Alex, Tamara, Paul, Philip, Maik, Sophie, Vasiliki, Johannes, Felix, Wenqi, and Florian, you guys are an absolute pleasure to be surrounded by, scientifically and emotionally.

A special thanks to my fellow TRABITers. We have shared an incredible opportunity to collaborate, thrive, and learn from each other. Lucas, Eze, Daniel, Andrey, Athena, Amnah, Luca, Ines, Stefano, Thomas, Sveinn, Francesco, and Carmen, my

deepest appreciation is for you guys.

I would like to acknowledge my senior scientific collaborators, who have extended their supportive hand whenever I needed the most. I would like to express my heartfelt gratitude to Nils for guiding me in my first-ever Ph.D. project and heading the thesis committee, Uli for concretizing a vague idea with mathematical rigor and ongoing collaborations, and George and Volker for effortlessly joining the forces during my concluding Ph.D. project. A special thanks to Jan, Bene, and Dennis for enriching me with their intermittent dosage of clinical expertise.

I express my gratitude to the TRABIT and the DCoMEX for supporting me financially throughout my doctoral studies and Graduate Center for Bioengineering for hosting me with open arms.

I would like to gratefully acknowledge Prof. Nils Forkert and Prof. Bjoern Andres for agreeing to examine my thesis on a very short notice.

A significant source of emotional support far from home comes from my dearest friends outside academic premises, who became indispensable during the pandemic. Thank you, Rajat, who also became one of my critical research partners a few months down the line, Poulami, Anindya, Abhi, Rahul, Soumya, Susnata, Sovan, Anwesha, and Punam, for all the nonsense talk and precious time spent. That was essential to keeping me sane, emotionally healthy, and motivated throughout.

Last but not least, I am most in debt to my parents, whose sacrifice gave me the wings to travel across continents and find my own way. I know you miss me a lot for not seeing me for a long time. You give me strength every day and are the source of my inspiration and dream. I miss you too. Love and respect.

Contents

List of Figures

Publication List

The following four publications constitute the core of my *cumulative doctoral thesis*. A * indicates shared first authorship.

[1] **S. Shit**, D. Das, I. Ezhov, J. C. Paetzold, A. F. Sanches, N. Thuerey, and B. H. Menze. "Velocity-To-Pressure (V2P)-Net: Inferring Relative Pressures from Time-Varying 3D Fluid Flow Velocities." In: *Proceedings of the International Conference on Information Processing in Medical Imaging*. Springer. 2021, pp. 545–558.

[2] **S. Shit**, J. Zimmermann, I. Ezhov, J. Paetzold, A. F. Sanches, C. Pirkl, and B. H. Menze. "SRflow: Deep Learning Based Super Resolution of 4D-flow MRI Data." In: *Frontiers in Artificial Intelligence* (2022).

[3] **S. Shit***, J. C. Paetzold*, A. Sekuboyina, I. Ezhov, A. Unger, A. Zhylka, J. P. Pluim, U. Bauer, and B. H. Menze. "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, pp. 16560–16569.

[4] **S. Shit***, R. Koner*, B. Wittmann, J. Paetzold, I. Ezhov, H. Li, J. Pan, S. Sharifzadeh, G. Kaissis, V. Tresp, and B. H. Menze. "Relationformer: A Unified Framework for Image-to-Graph Generation." In: *Proceedings of the European Conference on Computer Vision*. Springer Nature, 2022, pp. 422–439.

The following additional 24 publications were further co-authored *during the time of my doctoral thesis*. A ***** indicates shared first authorship.

2022

[1] N. Stucki*, J. C. Paetzold*, **S. Shit***, B. Menze, and U. Bauer. "Topologically faithful image segmentation via induced matching of persistence barcodes." In: *arXiv preprint arXiv:2211.15272* (2022).

[2] B. Wittmann, F. Navarro, **S. Shit**, and B. Menze. "Focused Decoding Enables 3D Anatomical Detection by Transformers." In: *arXiv preprint arXiv:2207.10774* (2022).

[3] B. Wittmann, **S. Shit**, F. Navarro, J. C. Peeken, S. E. Combs, and B. H. Menze. "SwinFPN: Leveraging Vision Transformers for 3D Organs-At-Risk Detection." In: *Medical Imaging with Deep Learning*. 2022.

[4] **S. Shit**, C. Prabhakar, J. C. Paetzold, M. J. Menten, B. Wittmann, I. Ezhov, et al. "Graflow: Neural Blood Flow Solver for Vascular Graph." In: *Geometric Deep Learning in Medical Image Analysis (Extended abstracts)*. 2022.

[5] I. Ezhov, K. Scibilia, K. Franitza, F. Steinbauer, **S. Shit**, L. Zimmer, J. Lipkova, F. Kofler, J. C. Paetzold, L. Canalini, et al. "Learn-Morph-Infer: a new way of solving the inverse problem for brain tumor modeling." In: *Medical Image Analysis* 83 (2023), p. 102672.

[6] I. Ezhov, M. Rosier, L. Zimmer, F. Kofler, **S. Shit**, J. C. Paetzold, K. Scibilia, F. Steinbauer, L. Maechler, K. Franitza, et al. "A for-loop is all you need. For solving the inverse problem in the case of personalized tumor growth modeling." In: *Machine Learning for Health*. PMLR. 2022, pp. 566–577.

[7] F. Navarro, G. Sasahara, **S. Shit**, I. Ezhov, J. C. Peeken, S. E. Combs, and B. H. Menze. "A unified 3D framework for Organs at Risk Localization and Segmentation for Radiation Therapy Planning." In: *Proceedings of the 44th International Engineering in Medicine and Biology Conference (EMBC)*. 2022.

[8] F. Navarro, C. Watanabe, **S. Shit**, A. Sekuboyina, J. C. Peeken, S. E. Combs, and B. H. Menze. "Self-Supervised Pretext Tasks in Model Robustness & Generalizability: A Revisit from Medical Imaging Perspective." In: *Proceedings of the 44th International Engineering in Medicine and Biology Conference (EMBC)*. 2022.

[9] I. Horvath, J. Paetzold, O. Schoppe, R. Al-Maskari, I. Ezhov, **S. Shit**, H. Li, A. Ertuerk, and B. H. Menze. "METGAN: Generative Tumour Inpainting and Modality Synthesis in Light Sheet Microscopy." In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 227–237.

2021

[1] J. C. Paetzold, J. McGinnis, **S. Shit**, I. Ezhov, P. Büschl, C. Prabhakar, A. Sekuboyina, M. Todorov, G. Kaissis, A. Ertürk, and B. H. Menze. "Whole Brain Vessel Graphs: A Dataset and Benchmark for Graph Learning and Neuroscience." In: *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[2] K. Payette, P. de Dumast, H. Kebiri, I. Ezhov, J. C. Paetzold, **S. Shit**, A. Iqbal, R. Khan, R. Kottke, P. Grehten, et al. "An automatic multi-tissue human fetal brain segmentation benchmark using the Fetal Tissue Annotation Dataset." In: *Scientific Data* 8.1 (2021), pp. 1–14.

[3] **S. Shit***, I. Ezhov*, L. Mächler, J. Lipkova, J. C. Paetzold, F. Kofler, M. Piraud, and B. H. Menze. "Semi-Implicit Neural Solver for Time-dependent Partial Differential Equations." In: *arXiv preprint arXiv:2109.01467* (2021).

[4] H. Li, A. Menegaux, B. Schmitz-Koep, A. Neubauer, F. J. Bäuerlein, **S. Shit**, C. Sorg, B. H. Menze, and D. Hedderich. "Automated claustrum segmentation in human brain MRI using deep learning." In: *Human brain mapping* 42.18 (2021), pp. 5862–5872.

[5] L. Fidon, M. Aertsen, **S. Shit**, P. Demaerel, S. Ourselin, J. Deprest, and T. Vercauteren. "Partial supervision for the FeTA challenge 2021." In: *Workshop on Perinatal, Preterm and Paediatric Image Analysis*. 2021.

[6] L. Fidon*, **S. Shit***, I. Ezhov, J. C. Paetzold, S. Ourselin, and T. Vercauteren. "Generalized Wasserstein Dice Loss, Test-time Augmentation, and Transformers for the BraTS 2021 challenge." In: *Proceedings of the Brain Lesion (BrainLes) workshop*. 2021.

[7] L. Mächler, I. Ezhov, F. Kofler, **S. Shit**, J. C. Paetzold, T. Loehr, B. Wiestler, and B. H. Menze. "FedCostWAvg: A new averaging for better Federated Learning." In: *Proceedings of the Brain Lesion (BrainLes) workshop*. 2021.

[8] I. Ezhov, T. Mot, **S. Shit**, J. Lipkova, J. C. Paetzold, F. Kofler, C. Pellegrini, M. Kollovieh, F. Navarro, H. Li, et al. "Geometry-aware neural solver for fast Bayesian calibration of brain tumor models." In: *IEEE Transactions on Medical Imaging* (2021).

2020

[1] A. B. Qasim, I. Ezhov, **S. Shit**, O. Schoppe, J. C. Paetzold, A. Sekuboyina, F. Kofler, J. Lipkova, H. Li, and B. H. Menze. "Red-GAN: Attacking class imbalance via conditioned generation. Yet another medical imaging perspective." In: *Proceedings of the Medical Imaging with Deep Learning*. PMLR. 2020, pp. 655–668.

[2] M. I. Todorov, J. C. Paetzold, O. Schoppe, G. Tetteh, **S. Shit**, V. Efremov, K. Todorov-Völgyi, M. Düring, M. Dichgans, M. Piraud, et al. "Machine learning analysis of whole mouse brain vasculature." In: *Nature methods* 17.4 (2020), pp. 442–449.

[3] S. Gerl, J. C. Paetzold, H. He, I. Ezhov, **S. Shit**, F. Kofler, A. Bayat, G. Tetteh, V. Ntziachristos, and B. H. Menze. "A distance-based loss for smooth and continuous skin layer segmentation in optoacoustic images." In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 309–319.

[4] A. Bayat, **S. Shit**, A. Kilian, J. T. Liechtenstein, J. S. Kirschke, and B. H. Menze. "Cranial implant prediction using low-resolution 3D shape completion and high-resolution 2D refinement." In: *Proceedings of the Cranial Implant Design Challenge*. Springer. 2020, pp. 77–84.

[5] **S. Shit**, I. Ezhov, J. C. Paetzold, and B. H. Menze. "Aν-Net: Automatic Detection and Segmentation of Aneurysm." In: *Proceedings of the International workshop on Cerebral Aneurysm Detection*. Springer. 2020, pp. 51–57.

2019

[1] I. Ezhov, J. Lipkova, **S. Shit**, F. Kofler, N. Collomb, B. Lemasson, E. Barbier, and B. H. Menze. "Neural parameters estimation for brain tumor growth modeling." In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 787–795.

[2] F. Navarro*, **S. Shit***, I. Ezhov, J. Paetzold, A. Gafita, J. C. Peeken, S. E. Combs, and B. H. Menze. "Shape-aware complementary-task learning for multi-organ segmentation." In: *Proceedings of the International Workshop on Machine Learning in Medical Imaging*. Springer. 2019, pp. 620–627.

PART I

INTRODUCTION

Foreword

Deep learning has evolved rapidly in the last decade thanks to access to large datasets and increased computation capabilities. Specifically, supervised deep learning has emerged as an excellent tool for learning a parametric function of interest given enough input-output pairs. Nowadays, deep models' long reach finds usage ranging from analyzing small microscopic images [1] to big cosmological images [2]. At the core of this deep learning success lies the contribution of dedicated, modern, and powerful parametric models and well-engineered strategies to make these models learn up to their total capacity. However, most of this engineering effort was targeted at modalities like natural images [3] and natural languages [4] where semantic correspondence is pivotal to our understanding.

Carrying on its root success in computer vision and natural language processing, deep learning finds broader applications in healthcare, physics simulations [5], biology [6], and what not. Among these, the application of deep learning in healthcare and biology is of recurring interest in this thesis.

Now, a natural question arises: why do we need to pay special attention to deep learning applications in medical applications?

First, unlike machine learning applications in natural images and natural language processing, where *semantics* prevail as the unit of understanding information, the medical and biological field demands precise quantitative measurements without semantic ambiguity. The reason in most application scenarios is that the quantity of interest is linked to a physical entity, be it structural or functional. One example is Computed Tomography (CT) images, where the image intensity is directly linked to the tissue density. The same analogy goes for medical applications such as blood flow modelling, tumour growth modelling, and image-based biomarker identification. Hence, special measures must be developed to equip the machine learning models to obey the underlying biophysical constraints in order to be applicable to these fields. These additional requirements worsen the already existing difficulties, such as scarcity of domain knowledge and data privacy.

Second, a different level of abstraction is needed based on the scale and target

application we are interested in. For example, to solve the task of tubular structure segmentation in an image, one could quickly adopt a state-of-the-art convolutional model and an optimizer out-of-the-box from deep learning libraries like Pytorch or Tensorflow. However, the question of what to optimize is more puzzling. One could see that tubular structures form a dense prediction and argue for the correct prediction for each pixel. A second opinion is to shift the focus from local predictions to a more abstract concept of connectivity. A third view could be to go even higher and reduce the connectivity to a compact graph representation. A fourth abstraction could be adding different physical attributes to the graph, such as the radius for each edge, in order to recover the lost local details, and so on. These highlight the requirement of looking into the same problem from a different perspective. Each of them is a valid and well-reasoned approach; however, each approach requires finding suitable constraints to enforce the desired level of abstraction.

Keeping these two principles in mind, the key objective of this thesis is to explore new avenues in the methodological aspects of vascular image analysis. As mentioned before, vascular applications involve quantifying biophysical phenomena underlying an image measurement. Examples considered in this thesis are estimating pressure distribution from velocity measurement and super resolving an already measured velocity field. First, a specific strategy principle at the crossroads of the *physics constrained* and the neural network has to be investigated. Secondly, the representations of vascular objects are analyzed in this thesis, for which the underlying *topological and graph constraints* have to be taken into account for the deep models. The contributions of this thesis primarily revolve around incorporating the above-mentioned *physical constraints* into the modern deep models and their application in structural and functional analysis of image-based vascular applications. Among all possible elements of deep learning, the main focus of this dissertation will be on the network architecture and custom loss function.

Organization

This dissertation is arranged as follows. Part I consists of four chapters, of which the current one is Chapter 1. Chapter 2 introduces the application background and briefly familiarises the imaging modalities. The key concepts and methodology used in this dissertation, including a short literature review, are described in Chapter 3. Chapter 4 summarizes the main contributions of this thesis. Part II comprises four peer-reviewed publications, constituting the main contribution of this thesis in Chapters 5-8. Each journal and conference paper is presented in a self-contained section, starting with

a summary of the publication. In Part III of this thesis, Chapter 9 discusses the presented work and draw conclusion. Finally, in Chapter 10 an outlook on future research directions is given. Appendix A comprises supplementary material from three peer-reviewed publications and one peer-reviewer workshop publication that are not relevant to the evaluation of this thesis but complement the prominent publications thematically, either presenting supporting results or relevant side projects.

Background

In this chapter, the target vascular applications are sketched out to motivate the methodological contribution of this thesis. To complement the tasks hand, we will briefly go through two cutting-edge imaging modalities that have been considered in this thesis.

2.1 The Vascular System

A crucial jump in the evolutionary advantage was the circulatory system. A circulation for a continuous stream of nutrition, oxygen, sewage extraction, protection, and communication. For efficient transportation of such payloads, vessels have naturally emerged as structural networks for this transport. Notably, the structure of the vessel network is highly customized by the functional requirements of its end users. For example, in the brain, there has been extra redundancy called collateral in case of any abrupt disruption via one path. Similarly, the fine capillaries in the filter in the kidney are dedicated to minimizing the loss of valuable molecules and getting rid

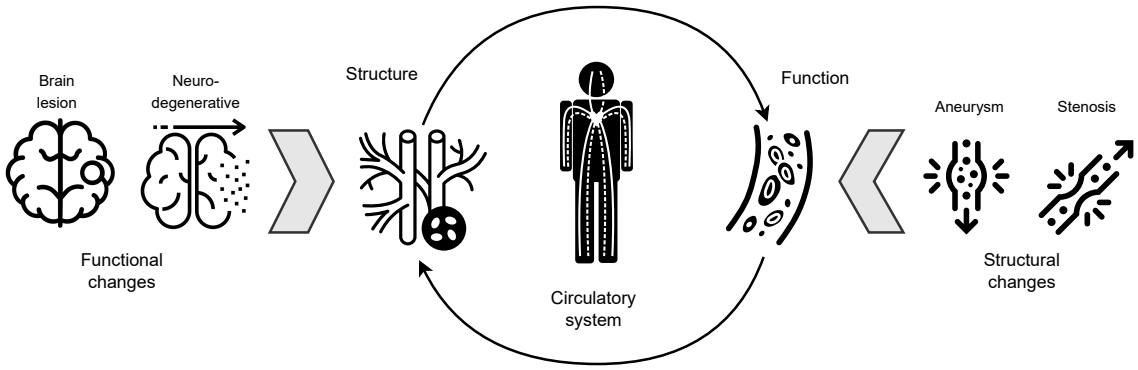

Figure 2.1: Inter-dependency of structural and functional properties of circulatory systems.

of harmful substances. Thus the structure of blood vessels heavily influences the functional part and any changes in the functions, in turn, change vessel structure (c.f. Fig 2.1). Below we briefly go through different applications of structural and functional quantities of vessel networks and blood flow.

2.1.1 Functional Vessel Analysis

Blood circulation is the core pillar for the proper functioning of each individual organ. To assess the quality of the current circulatory state, different functional measurement is part of the clinical routine, such as blood pressure and pulse rate. However, these measurements are a point estimator of the whole system. Recent advancement in non-invasive imaging, as discussed in the next section, opens up new possibilities for a novel way to measure and quantify hemodynamics even localized to a particular small region of interest. This is particularly useful to develop novel bio-marker from hemodynamic quantity, such as pressure distribution within brain aneurysms [7] to hemorrhagic stroke. Therefore, a major half of the thesis will build on new methods to tackle functional quantity estimation for vascular regions.

2.1.2 Structural Vessel Analysis

While functional analysis has enormous clinical importance, structural analysis of vascular networks also appears in many pre-clinical tasks. Thanks to our increasing understanding of optical engineering and biochemical compositions, we can probe into complete vasculature up to the capillary level. We all bear a typical genetic code that spawns a healthy vascular network. However, in the presence of different kinds of pathogens, the network deviates from its healthy condition. To efficiently understand this causal relationship, one needs to consider the different representations of the vascular network, especially emphasising its topological property and graph nature. The other half of the thesis devotes to developing tools for this purpose.

2.2 Vascular Imaging Modalities

Below, we will consider two prominent imaging modalities used in this thesis. 1) 4D-Flow MRI and 2) Light-sheet microscopy. Finally, we present a concise description relevant to understanding the imaging modalities, and enthusiastic readers are encouraged to consult associated references.

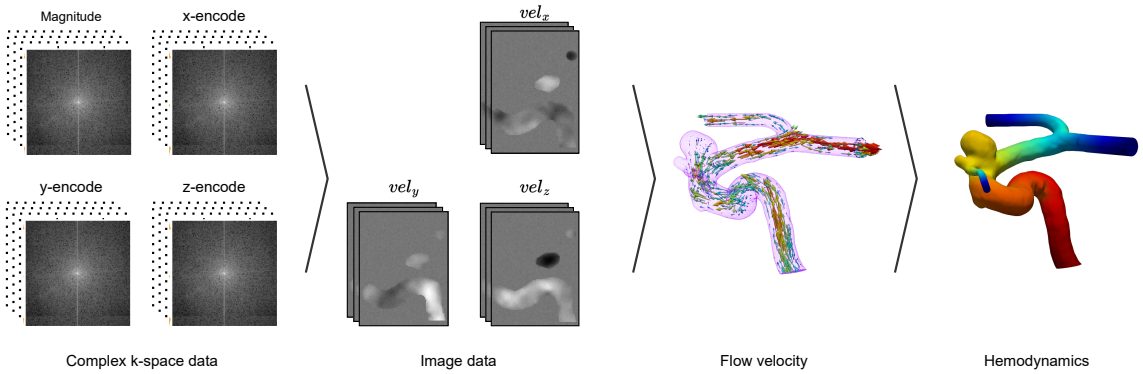

Figure 2.2: 4D-flow MRI processing pipeline: from measurement to quantitative hemodynamics analysis.

2.2.1 4D-Flow MRI

Out of many miraculous applications of quantum mechanics, Magnetic Resonance Imaging (MRI) stands out in the history of humanity. MRI equipped us with a high resolution *camera* to look inside the human body without harming the tissue or cell level. The core principle of MRI relies on the alignment of nuclear spin along the applied magnetic field and collects their signal through a radio-frequency transmitter-receiver. The raw MRI signals collect the complex signal in frequency space, known as k-space. The magnitude of the inverse Fourier reconstruction provides our desired image. Over time, a different protocol of imaging technique masters the need to well separate different tissues or organs through contrast. This includes T1-weighted, T2-weighted, Flair, and diffusion-weighted images.

Primarily, MRI images involve static anatomical delineation, and the measured intensity weakly correlates with the tissue density or any bio-physical quantities. Interestingly, it is also possible to measure quantitative signals through the phase of the complex MRI signal. The resultant modality is called phase-contrast MRI. In this setting, the motion of a moving spin is captured via differential computation in the phase component, which directly measures the fluid flow in our body, such as blood flow and cerebrospinal fluid flow (c.f. Fig 2.2). As a result, one can measure temporally resolved blood flow in a 3D volume. Hence, the name appears as 4D Flow MRI [8]. The following provides a brief outline of the imaging pipeline.

Phase-contrast MRI

The complex MRI signal is a function of spacial tissue properties, such as T_1 and T_2 relaxation time and the applied magnetic field. The phase of this complex signal can be expressed as

$$\phi(\mathbf{r}, t) = \gamma \Delta B T_E + \gamma \int_0^{T_E} \mathbf{G}(t) \cdot \mathbf{r}(t) \tag{2.1}$$

where $\mathbf{r}$ and t denotes space and time respectively, γ is gyromagnateic ratio, T_E is echo time, ΔB is sum of local inhomogeneity and $\mathbf{G}(t)$ is the gradient. $\mathbf{r}(t)$ can be expanded using Taylor series as $\mathbf{r}(t) = \mathbf{r}_0 + \mathbf{v}t + \frac{1}{2}\mathbf{a}t^2 + \ldots$, where $\mathbf{v}$ and $\mathbf{a}$ are velocity and acceleration respectively. Taking only first order tern and neglecting higher order, we get

$$\phi(\mathbf{r}, t) = \phi_0 + \gamma \mathbf{r}_0 \int_0^{T_E} \mathbf{G}(t)dt + \gamma \mathbf{v} \int_0^{T_E} \mathbf{G}(t)tdt. \tag{2.2}$$

First, one needs to set the $\mathbf{G}(t)$ such that the first integral becomes zero and the second integral is non-zero. To compensate for the ϕ_0, a second gradient needs to be applied where both the integral becomes zero. The difference can be represented as $\Delta\phi = \gamma \mathbf{v} M_1$. In other words, the velocity is a linear function of space $\mathbf{v} = \frac{\Delta\phi}{\gamma \mathbf{v} M_1}$. Because the maximum phase limit of $\pm\pi$, it leads us to velocity encoding (V_{enc}), which describes the measurement sensitivity towards the velocity magnitude. In order to measure velocities higher than V_{enc}, one needs to apply phase unwrapping.

In practice, because of the dynamic nature of the data, the measurement is synced with cardiac motion with ECG triggering to denote the beginning of a cardiac cycle and the time series is filled in over more than one cardiac cycle. Further, to enable 3D measurement, an additional two gradient is needed to nullify unwanted signals effectively. In recent times, many breakthroughs accelerated the acquisition to enable 4D-Flow MRI feasibly.

Image Based Hemodynamics

Based on the acquired 4D signal, one can estimate different hemodynamic quantities of interest [8]. First, stage the region of interest is identified in the image by segmentation of contouring. Next, given the velocity measurement, one has to compute the relevant biophysical quantities such as pressure and stress tensor of wall shear stress. These quantities are found to be important bio-marker of different cardiovascular and neurovascular diseases. Further, given the signal-to-noise ratio of

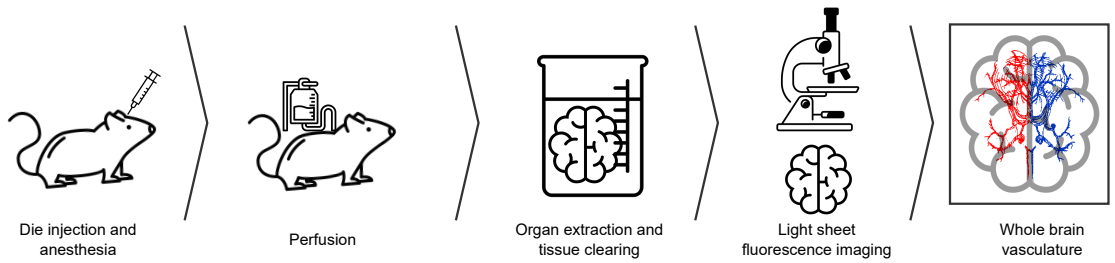

Figure 2.3: Light sheet microscopy processing pipeline: from biochemical preparation to image acquisition.

the MRI signal, stable spatial differentiation reacquires substantial resolution, which is a super-resolution problem.

2.2.2 Light Sheet Microscopy

In order to image organs in a traditional histopathological microscopy setting, one needs to dissect the organ into thin slices which are then imaged. Finally, these images have to be stitched together in 3D. In contrast, light sheet microscopy technology allows us to scan a whole organ or even an entire specimen in 3D. This is a significant advantage compared to previous methods where the slicing and stitching of tissue introduced substantial errors and artefacts. However, for that purpose, the specimen must be transparent, and the objects of interest must be highlighted with a fluorescent dye. A recent technological breakthrough in tissue-clearing [9] enables taking full advantage of light sheet microscopy to scan biological specimens in full 3D. The processing pipeline is shown in Fig. 2.3. The protocol involves selecting the appropriate fluorescence dye and fine-tuning the clearing protocol to the particular tissues and anatomy of interest.

Subsequently, the organ is placed under the microscope to scan a fully volumetric 3D scan keeping the organs intact. In light sheet microscopy, the illumination is done in a single plane instead of the whole 3D volume. This setup reduces the background noise as much as possible and scans through the whole volume slice by slice.

Methodology

As discussed in the previous chapters, this thesis aims to explore the desired physical constraints to equip the machine learning model to deal with the problem at hand. In this thesis, I will mainly look into three cases in the neurovascular application, a) flow computation, b) topology preservation in a vascular graph, and b) vascular graph extraction. In addition, we will be looking into the existing prior art and finding room for improvement to set the problem statements for the thesis.

Before jumping into specific aspects of the deep learning-based literature, which pays special attention to retaining the link between model prediction and the physical entity of interest, We will briefly review the building blocks of modern deep learning.

3.1 Building Blocks of Deep Learning

A decade has passed since the spring of deep learning initiated by AlexNet [10] and the field since then have been matured and standardized. The main components of modern deep learning can be boiled down into two parts; a) designing a search space, e.g., model architecture and b) designing a search strategy, which includes loss functions, regularizer, optimizer, and scheduler. Different engineering directions toward better search space and search-strategy have matured and benefited from shared and complementary design principles. The focus on improving the search space, a.k.a model, has gifted us ResNet, DenseNet, U-Net, and most recently, Transformer. Simultaneous exploration of loss functions, optimizer, and regularization, have boosted performance and accelerated train time.

The deep learning model can be expressed as a parametric model $\mathcal{F}_\theta^{\mathcal{A}}$ with d dimensional learnable parameter $\theta \in \mathbb{R}^d$ and architecture $\mathcal{A}$. The architecture helps to find a meaningful prediction from the search space in $\mathbb{R}^d$. The learning problem can be formulated as finding the right set of parameter θ given a criterion $\mathcal{L}$ and a dataset $\mathcal{D} = \{\boldsymbol{x}, \boldsymbol{y}\}$

$$\theta_{\text{opt}} = \arg\min_\theta \mathcal{L}(\mathcal{F}_\theta^{\mathcal{A}}(\boldsymbol{x}), \boldsymbol{y}) \tag{3.1}$$

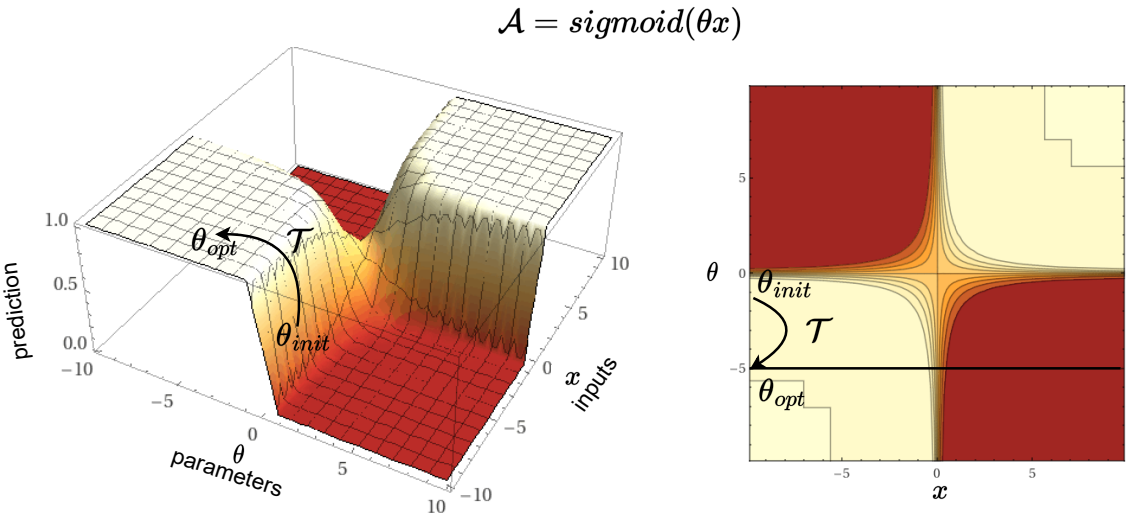

Figure 3.1: A pictorial description delineating training strategy (τ) from θ_{init} to θ_{opt} under an architecture $\mathcal{A}$.

The training part of the learning is a process of mapping an initial guess θ_{init} to an optimal value θ_{opt} of the parameter. Hence, a training strategy defined by the following mapping

$$\mathcal{T} : \theta_{\text{init}} \rightarrow \theta_{\text{opt}}; \text{ given } \mathcal{L} \tag{3.2}$$

With the optimal θ_{opt}, the inference $\boldsymbol{y}$ for an input sample $\boldsymbol{x}$ is obtained

$$\mathcal{F}^{\mathcal{A}}_{\theta_{\mathbf{opt}}} : \boldsymbol{x} \rightarrow \boldsymbol{y} \tag{3.3}$$

Together with carefully engineered search space ($\mathcal{F}^{\mathcal{A}}_{\theta}$) and search strategy ($\mathcal{T}, \mathcal{L}$) have contributed to the immense success of deep learning in the past decade. A primary goal of this thesis is to incorporate various physical constraints such as partial differential equation, flow consistency, underlying topology, or graph structure into the architecture and the loss function. Before delving deep into each of the individual topics, the following is a very brief overview of modern deep learning model components, namely architectures ($\mathcal{A}$) and loss functions ($\mathcal{L}$).

3.1.1 Model Architecture

Different model architectures have been developed, keeping specific applications in mind. The architecture has many vital components, such as the functional layer, norm

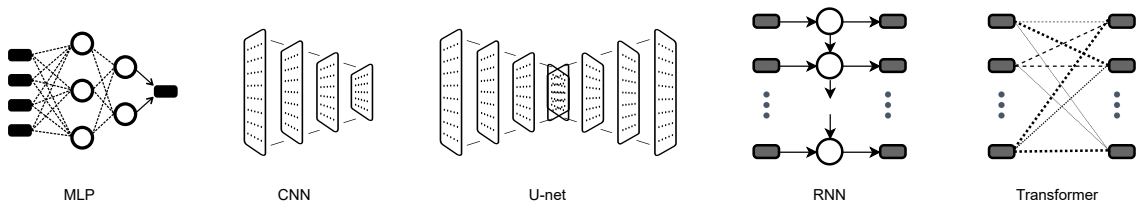

Figure 3.2: Pictorial comparison of different deep learning architectures.

layer, an activation layer. Here we will focus solely on the functional layer. Functional layers are designed to specifically take advantage of invariance or equivariance of the input space under certain spatio-temporal transformations. This is often referred to as imposing inductive biases in the literature.

MLP

A Multi-Layered Perceptron (MLP) [11] is a fully-connected feed-forward neural network. A single layer of an MLP consists of a weight matrix ($\boldsymbol{W}$) and a bias vector ($\boldsymbol{b}$). A two-layer MLP can be described below

$$\mathcal{F}_{MLP}(\boldsymbol{x}) = \boldsymbol{W}_2(\sigma(\boldsymbol{W}_1(\boldsymbol{x}) + \boldsymbol{b}_1)) + \boldsymbol{b}_2 \tag{3.4}$$

where σ is an activation function.

CNN

Unlike MLP, a Convolutional Neural Network (CNN) [12] relies on convolution with a set of filters ($\boldsymbol{W}$). Convolutional layers are stacked together to form a CNN. A two-layer CNN can be described as follows

$$\mathcal{F}_{CNN}(\boldsymbol{x}) = \boldsymbol{W}_2 \circledast (\sigma(\boldsymbol{W}_1 \circledast \boldsymbol{x} + \boldsymbol{b}_1)) + \boldsymbol{b}_2 \tag{3.5}$$

Additionally, the CNN network consists of downsampling layers to reduce the spatial dimension of the image as the feature size grows with layers.

U-net

MLP and CNN, as perceived initially, was suitable classification task. Over time, U-net [13] has been developed based on CNN to make dense predictions on the image.

In addition to the downsampling layer, U-net relies on an upsampling layer to rescale feature maps into the image dimension. A U-net of depth two can be expressed as.

$$\begin{aligned} \boldsymbol{y}_1 &= \boldsymbol{W}_1 \circledast \boldsymbol{x} + \boldsymbol{b}_1 \\ \boldsymbol{y}_2 &= \boldsymbol{W}_2 \circledast \sigma(\boldsymbol{y}_1) + \boldsymbol{b}_2 \\ \mathcal{F}_{U-net}(\boldsymbol{x}) &= \boldsymbol{W}_4 \circledast (\sigma(\boldsymbol{W}_3 \circledast \boldsymbol{y}_2 + \boldsymbol{b}_3) \oplus \boldsymbol{y}_1) + \boldsymbol{b}_4 \end{aligned} \tag{3.6}$$

RNN

A Recurrent Neural Network (RNN) [14]was introduced to process sequence data such as text and speech. The past input is processed and stored in a hidden state ($\boldsymbol{h}$), which is combined with the current input to produce the output.

$$\boldsymbol{h}_t = \mathcal{F}_h(\boldsymbol{h}_{t-1}, \boldsymbol{x}_{t-1}) \tag{3.7}$$

$$\boldsymbol{y}_t = \mathcal{F}_{RNN}(\boldsymbol{h}_t, \boldsymbol{x}_t) \tag{3.8}$$

Transformer

RNN suffers from vanishing gradient and modeling long-range dependency. Hence a new variation of the model has evolved based on the attention mechanism [15]. The attention mechanism works on a sequence data ($\boldsymbol{x} = \{\boldsymbol{x}_t\}_{t=1}^{T}$)

$$\begin{aligned} \boldsymbol{q} = MLP_q(\boldsymbol{x}), \boldsymbol{k} = MLP_k(\boldsymbol{x}), \boldsymbol{v} = MLP_v(\boldsymbol{x}) \\ Attn(\boldsymbol{x}) = softmax\left(\frac{\boldsymbol{q}\boldsymbol{k}^\top}{\sqrt{d}}\right)\boldsymbol{v} \end{aligned} \tag{3.9}$$

multiple heads with attention mechanisms make a strong model for modern machine-learning tasks.

3.1.2 Loss Function

Similar to the architectural revolution, deep learning heavily progressed with the novel design of loss functions. The loss functions have come from diverse communities dealing with a spectrum of problems. Here we will categorize them broadly into a few buckets for the ease of revisiting them. We will consider the ground truth as $\boldsymbol{y}$ and the prediction as $\hat{\boldsymbol{y}}$.

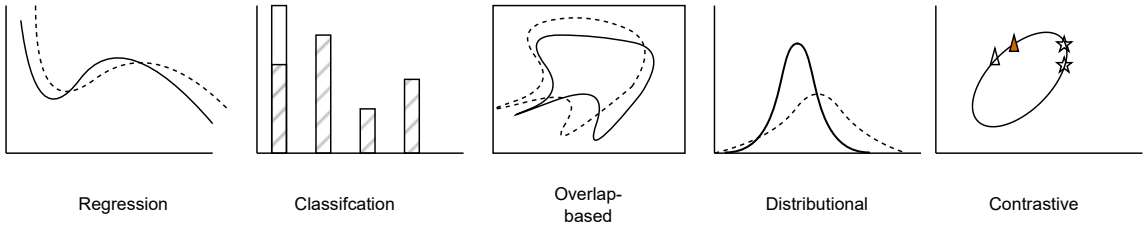

Figure 3.3: Pictorial comparison of the commonly used loss function.

Regression

Regression loss is used for predicting real-valued numbers. It uses ℓ_p norm as a loss function with a usual value of $p = 1, 2$. ℓ_1 loss is robust compared to the ℓ_2 loss. The general regression loss is given below.

$$\mathcal{L}_{reg} = \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|_p \tag{3.10}$$

There are numerous varieties of regression loss apart from ℓ_P norm, including Huber loss etc.

Classification

Unlike regression, classification loss involves computing loss with categorical variables. The commonly used loss function is cross-entropy

$$\mathcal{L}_{CE} = -\boldsymbol{y} \log(\hat{\boldsymbol{y}}) - (1 - \boldsymbol{y}) \log(1 - \hat{\boldsymbol{y}}) \tag{3.11}$$

Over time many different classification losses have been proposed, like focal loss, hinge loss, etc.

Overlap-based

Unlike pointwise classification, there is a higher-order abstraction of objects for which overlap-based loss has been used. Among them, the IoU of Dice [16] is a popular loss used in object detection and segmentation.

$$\mathcal{L}_{Dice} = \frac{\sum(\boldsymbol{y} \odot \hat{\boldsymbol{y}})}{\sum \boldsymbol{y} + \sum \hat{\boldsymbol{y}}} \tag{3.12}$$

Distributional

A different kind of loss, namely, distributional loss, has dominated the field of generative modelling in the recent past. The notion is derived from optimal transport or one distribution to a different one [17]. The main distributional loss is Wasserstein distance. p-wasserstein is defined as

$$\mathcal{L}_W = \inf_{\Pi} \left\{ \left(\mathbb{E}_{(\hat{\boldsymbol{y}},\boldsymbol{y})\sim\Pi}[d(\hat{\boldsymbol{y}},\boldsymbol{y})^p] \right)^{\frac{1}{p}} \right\} \tag{3.13}$$

Contrastive

Recently a very useful class of loss functions came into use. Contrastive loss [18], at the same time, tries to maximize the similarities of positive samples from an anchor and maximize the distance for negative samples. There are many flavours of contrastive learning, self-supervised and supervised. The supervised contrastive loss [19] is

$$\mathcal{L}_C = \frac{-1}{|P(i)|} \sum_{p\in P(i)} \log \frac{\exp\left(\hat{\boldsymbol{y}}_i \cdot \hat{\boldsymbol{y}}_p/\tau\right)}{\sum_{a\in A(i)} \exp\left(\hat{\boldsymbol{y}}_i \cdot \hat{\boldsymbol{y}}_a/\tau\right)} \tag{3.14}$$

where $P(i) \equiv \left\{p \in A(i) : \boldsymbol{y}_p = \boldsymbol{y}_i\right\}$ is the set of indices of all positives with respect to anchor i and $A(i) \equiv \mathcal{D}\backslash\{i\}$

3.2 Physics Constrained Deep Learning

As discussed at the beginning of this chapter, one core problem is examining blood flow modelling through the machine learning lens. One has to solve a set of Partial Differential Equations (PDE), the Navier-Stokes equation, in this case, to model the flow behaviour. A recent surge in the crossroads of PDE and neural networks co-evolved two main research fields. 1. Taking inspiration from classical neural solver to improve deep learning architecture [20]; and 2. Incorporating modern deep models to design numerical solver [21]. The first category became immensely popular for modelling finite transformation from one set of inputs to another, thereby solving classification and shape modelling. Nevertheless, computer vision problems, strong constraints on search space, or the solution is hard to estimate beforehand. However, in the case of solving PDE, these constraints come free of cost as the form of governing equation, initial condition, and boundary condition.

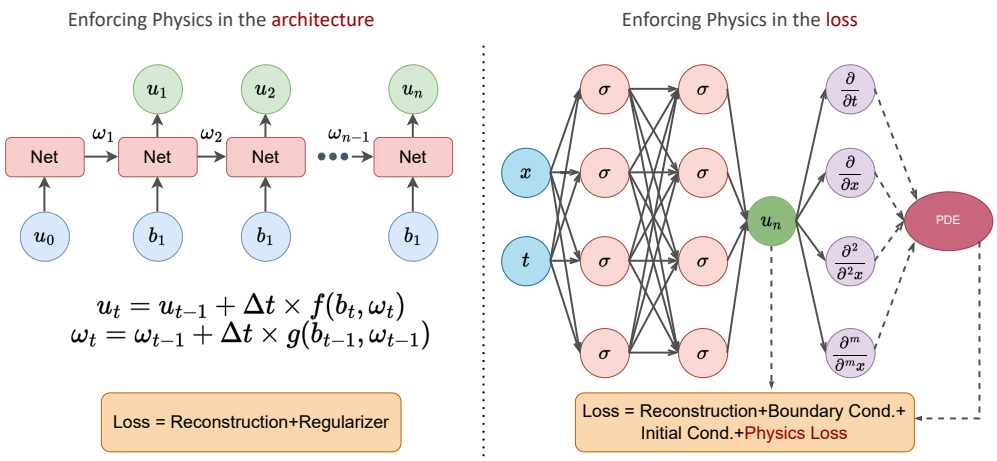

Figure 3.4: Physics constraints in the architecture vs physics constraints in the loss.

Again, two distinct design principles emerge as de-facto models in this avenue. They involve imposing the physics constrained in the search space (i.e. in the architecture) or the search strategy (i.e. in the loss). Let us consider the following PDE equation

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} &= \mathcal{F}(\boldsymbol{u}; \{\partial_i, \boldsymbol{a}_i\}_{i=1:N}), \forall x \in \Omega, \\ \text{s.t. } \mathcal{G}(\boldsymbol{u}) &= \boldsymbol{b}(x,t), \forall x \in \Gamma \text{ and } \boldsymbol{u}(x,t_0) = \boldsymbol{u}_0 \end{aligned} \tag{3.15}$$

where Ω is the domain (e.g., $\subseteq \mathbb{R}^2$ or $\subseteq \mathbb{R}^3$), Γ is the boundary with boundary function $\mathcal{G}$ and boundary values $\boldsymbol{b}(x,t)$ and $\boldsymbol{u}_0$ the initial value at time $t = t_0$. $\mathcal{F}(\boldsymbol{u}, \{\partial_i, \boldsymbol{a}_i\}_{i=1:N})$ is a spatial function comprising partial differential operators $\{\partial_i\}_{i=1:N}$ and its corresponding parameter $\{\boldsymbol{a}_i\}_{i=1:N}$. One can take one of the following roads to solve this PDE.

3.2.1 Physics in the Architecture

Most of the PDEs involve time as a variable. In practice, due to the intractability of the close-form solution, spatio-temporal discretization is employed. While spatial discretization is analogous to extracting image features, temporal discretization gives rise to a recurrent solution scheme [22, 23], which is analogous to recurrent neural networks. This connection can be leveraged as a constraint on the network architecture. Precisely, the recurrence scheme can be unfolded as RNN, and particular

temporal discretization can be imposed in the connection [24, 23]. One can write the following discretization scheme of Eq. 3.15.

$$u_{t+\Delta t} = u_t + \Delta t F(u_t; \mathbb{D}, \mathbb{A}) \tag{3.16}$$

To mimic Eq. 3.16, one can employ the following RNN with constrained architecture $\boldsymbol{f}$ and $\boldsymbol{g}$.

$$\begin{aligned} u_t &= u_{t-1} + \Delta t f\left(b_t, \omega_t\right) \\ \omega_t &= \omega_{t-1} + \Delta t g\left(b_{t-1}, \omega_{t-1}\right) \end{aligned} \tag{3.17}$$

The advantages of incorporating constraints in the architecture are that they can be easily generalized to different computation domains or geometry. On the other hand, because of particular temporal discretization, one has to take care of temporal stability by checking the time step.

3.2.2 Physics in the Loss

As an alternative approach, solving PDE can be thought of as an implicit function learning which maps the space and time to a variable of interest through a parametric model. To resolve the ambiguity one has to enforce the physics constrained in the loss of the model [25]. The resultant solution obeys the governing equation nearly. For Eq. 3.15 one can learn a parametric function $\mathcal{F}_\theta$

$$u = \mathcal{F}_\theta(x, t) \tag{3.18}$$

Subsequently, one can exploit the end-to-end differentiability to cheaply compute the spatial and temporal derivative of u. The resultant entities are put together in the loss to obey the governing equation ($\mathcal{L}_{\text{PDE}}$), initial ($\mathcal{L}_{\text{init}}$), and boundary condition ($\mathcal{L}_{\text{bound}}$).

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{init}} + \mathcal{L}_{\text{bound}} \tag{3.19}$$

where

$$\mathcal{L}_{\text{PDE}} = \mathcal{L}\left(\frac{\partial \mathcal{F}_\theta(x,t)}{\partial t}, \mathcal{F}(\mathcal{F}_\theta(x,t); \{\partial_i, \alpha_i\}_{i=1:N})\right);\ \forall x \in \Omega, t > t_0 \tag{3.20}$$

$$\mathcal{L}_{\text{init}} = \mathcal{L}(\mathcal{F}_\theta(x,t), u(x,t_0));\ \forall x \in \Omega, t = t_0 \tag{3.21}$$

$$\mathcal{L}_{\text{bound}} = \mathcal{L}(\mathcal{F}_\theta(x,t), \mathcal{B}(x,t));\ \forall x \in \Gamma, t > t_0 \tag{3.22}$$

Because of the implicit function learning, the network can handle continuous spatio-temporal interpolation. However, this kind of model suffers from poor generalizability and has to be retrained on every new geometry from scratch.

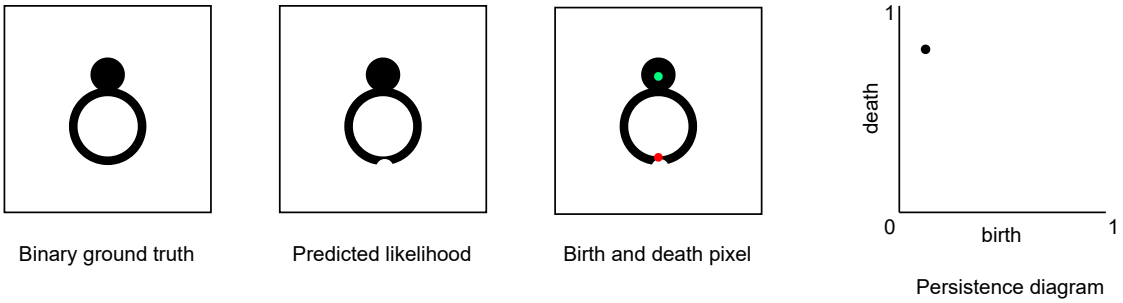

Figure 3.5: Persistence homology allows us to compute critical points in the predicted likelihood by checking all real values and identifying points where a topological entity, such as cycles, is born and dies.

3.3 Topology Preserving Image Segmentation

An essential recurring notion highly relevant at the crossroads of machine learning and physical systems is predicting the correct topology of an entity of interest. One example is the blood vessel segmentation task, where the segmented objects with incorrect topology (connectivity) would mean completely different functional properties, such as blood flow. All our work in this thesis is restricted to 2D and 3D, whilst the mathematical concepts of topology go beyond this.

A significant drawback of convnet-based prediction is that it outputs a prediction on a regular volumetric grid. Any topological computation beyond the voxel-wise metric requires establishing higher-order entities such as edge, face, and cubical complexes. A major bottleneck is that this is possible only in the presence of thresholded volume, which, if applied directly, would break end-to-end differentiability. Persistence homology allows us to circumvent this problem and find out the critical location [26] in an image for all possible discrete thresholds present in that image. The critical locations are essential in maintaining the discovered topological entities in the predicted likelihood. In the loss, the critical pixels are given special care to have a correct prediction. However, this demands heavy computation power since identifying the cycles can not be efficiently parallelized. One could question the necessity of spending tremendous computational power against finding only a sparse collection of critical pixels.

In summary, the aim of tackling the topological correctness problem is to find critical pixels, which, if having a predicted value lower/higher than the threshold, would

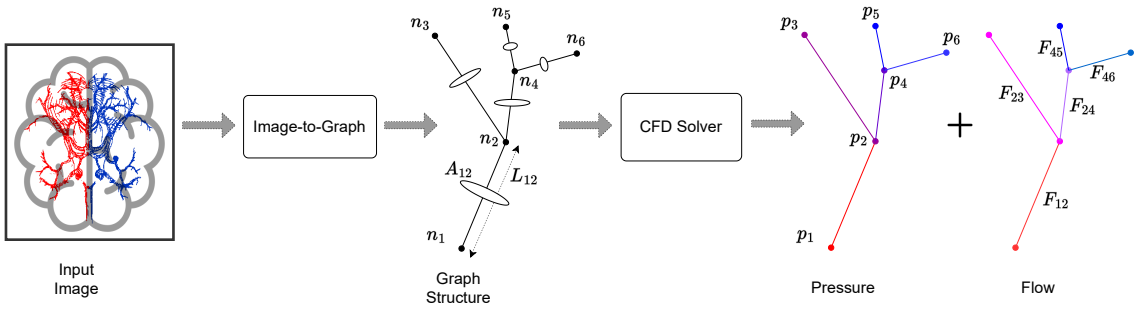

Figure 3.6: Capillary level vessel graph extraction is a prerequisite for flow-based bio-marker identification.

lead to the loss/extra of topological entities in the binarized image. Subsequently, during the training, the critical pixels should be emphasized. Now, can we find the critical pixels efficiently if we relaxed the criteria to include an increased number of pixels to be highlighted?

3.4 Image-to-Graph Generation

In many applications, segmentation is an intermediary step to the final representation. In the case of the blood vessel, the final representation of interest is graph structure [27]. Although the graph is a reduced representation than segmentation, this is an effective strategy for a variety of blood vessels, except the big artery aorta and heart. The resultant graph structure is used to analyze structural and functional properties. Among structural properties, geometry and thickness change of angle is of interest to study. Among functional properties, simulating blood flow in graphs and computing hemodynamic quantities are of paramount interest in many applications such as Alzheimer's or stroke.

Classically, one can extract centerlines from segmentation and extract a graph from it. However, since the skeleton would result in a voxelized binary representation, it would be dense and would have filled with a vast amount of unnecessary nodes of degree two. One can attempt to prune it, which, however, involves complex heuristics. An alternative direction is to fit spheres of small radii inside the segmentation and stitch the centroids to get centerlines. Although this results in a relatively smoother graph, the process requires substantial computation power. One can ask if we can learn a model to learn where to place nodes in an image and also learns to connect

them with appropriate edges.

Notably, graph extraction from an image is a relatively new field of research and mainly involves semantic knowledge graphs, i.e., scene-graph extraction from natural images [28]. Can we merge these two parallel endeavours under a single framework? Moreover, can we benefit from translating image-to-graph models from the scene-graph community to structural graph extraction?

Summary of Contributions

Having presented the scope of the application and relevant literature review in the previous chapter, here I present the concrete problem statement and the proposed methodology. The recurring theme of this dissertation is to enforce the known apriori constraints on the solution space.

For that, we will first consider the functional analysis aspects of vascular analysis and ask the following question.

Q1. Can we leverage a partial differential equation in the neural network for pressure estimation from velocity data? If so, what could be an efficient strategy to realize it?

Pressure distribution is an essential biomarker for different vascular pathogens. Estimating pressure from velocity measurement requires solving Poisson's equation with the Neumann boundary condition. Pressure gradient originates in two forms. First, the local changes in geometry influence advection and, thereby, a local gradient. Second, the transient nature at the boundary induces a global distribution. Based on this observation, we model two components with two different models. The first one is handled by a fully convolutional network with physics constrained in the architecture. The latter is tackled by an implicit neural function trained with boundary-driven loss. The fully convolutional network can be easily generalized over different geometry. However, the implicit function needs to be optimized separately for each new case. The proposed approach is validated with the help of simulated ground truth. Chapter-5 presents this work in detail.

Next, we look into the task of super-resolving the measured flow field and ask the following question.

Q2. Can we impose directional awareness in the 3D velocity field super-resolution, and what will be the benefits?

Increasing the resolution of the 4D-Flow MRI is necessary for the downstream

analysis. Traditional regression loss used for super-resolution tasks does not take any prior knowledge of the variable of interest into account. Primarily because of the Gaussian/Laplacian assumption on the noise distribution of each component, we use mean squared/mean absolute error for losses. However, its direction and magnitude are of immense interest for velocity. Separately enforcing its directionality is challenging because of the different ranges of the error bound for magnitude and its angle. Alternatively, we aim to leverage the projection of prediction onto ground truth and vice-versa to promote directional sensitivity to the loss function. We studied in residual learning settings with simulated and real data, which showed improved reconstruction than previous non-deep learning methods. Chapter-6 describes the peer-reviewed journal on this topic.

Next, we look into the structural aspects of the vascular network and focus on the segmentation task of vasculature from images and look into the following question.

Q3. How to enforce topological correctness in the form of connectivity on the segmentation, and what would be an efficient strategy to enforce this constraint?

Estimating the correct topological structure for the vessel graph is crucial. However, computing the topological features on a real-valued likelihood map is challenging without breaking end-to-end differentiability. Conventional use of persistence homology looks for a very sparse set of important points and emphasizes them in loss, however, at the cost of an enormous computation. Given the scale of the image size, the choice of important points selection can be flexible, which will drastically speed up the computation. We propose including the predicted skeleton on its ground truth and vice versa as a proxy to compare topology with a provable guarantee. Deriving from this principle, a differentiable loss is proposed comprising max-pooling and ReLU. Our experiments suggested improved topological results in 2D and 3D vessel segmentation tasks. Chapter-7 describes the peer-reviewed conference article on this topic.

Finally, we look into direct graph prediction from images without an explicit segmentation stage.

Q4. Can we put graph constraints on the network architecture to directly infer the underlying graph representation? If so, can we leverage techniques for other similar problems, and how far can we go?

A single-stage transformation from image representation to graph representation

is challenging yet important in the structural analysis of vascular networks. An image lies in a regular lattice structure, while the underlying graph lies in an unstructured set. We propose leveraging the transformer's cross-attention to bridge between two representations. This allows efficient joint learning of object and relation representation. The resulting representation not only learns where to place the node but also connects them based on a similarity measure whenever there should be an edge between them. We provided the first learning-based solution for 3D image-to-graph extraction and showed a proof-of-concept application where the ground truth graph is known. The final Chapter-8 consists of the peer-reviewed conference article on this problem.

PART II

PUBLICATIONS

Velocity-To-Pressure (V2P) - Net: Inferring Relative Pressures from Time-Varying 3D Fluid Flow Velocities

Suprosanna Shit, Dhritiman Das, Ivan Ezhov, Johannes C. Paetzold, Augusto Fava Sanches, Nils Thuerey & Bjoern H. Menze

Conference: International Conference on Information Processing in Medical Imaging (IPMI), June 2021

Synopsis: Pressure inference from a series of velocity fields is a common problem arising in medical imaging when analyzing 4D data. Traditional approaches primarily rely on a numerical scheme to solve the pressure-Poisson equation to obtain a dense pressure inference. This involves heavy expert intervention at each stage and requires significant computational resources. Concurrently, the application of current machine learning algorithms for solving partial differential equations is limited to domains with simple boundary conditions. We address these challenges in this paper and present V2P-Net: a novel, neural-network-based approach as an alternative method for inferring pressure from the observed velocity fields. We design an end-to-end hybrid-network architecture motivated by the conventional Navier-Stokes solver, which encapsulates the complex boundary conditions. It achieves accurate pressure estimation compared to the reference numerical solver for simulated flow data in multiple complex geometries of human in-vivo vessels.

Contributions of thesis author: Conceptualized the project, gathered necessary software resource, developed the framework to combine convnet and neural implicit functions, performed a leading role in the experiment and writing the manuscript.

5. Velocity-To-Pressure (V2P) - Net: Inferring Relative Pressures from Time-Varying 3D Fluid Flow Velocities

Velocity-To-Pressure (V2P) - Net: Inferring Relative Pressures from Time-Varying 3D Fluid Flow Velocities

Suprosanna Shit[1( )], Dhritiman Das[1,2], Ivan Ezhov[1], Johannes C. Paetzold[1], Augusto F. Sanches[1,3], Nils Thuerey[1], and Bjoern H. Menze[1,4]

[1] Department of Informatics, Technical University of Munich, Munich, Germany
{suprosanna.shit,ivan.ezhov,johannes.paetzold,nils.thuerey}@tum.de
[2] McGovern Institute for Brain Research, MIT, Cambridge, MA, USA
dasd@mit.edu
[3] Institute of Neuroradiology, University Hospital LMU, Munich, Germany
augusto.sanches@med.uni-muenchen.de
[4] Department of Quantitative Biomedicine, University of Zurich, Zürich, Switzerland
bjoern.menze@uzh.ch

Abstract. Pressure inference from a series of velocity fields is a common problem arising in medical imaging when analyzing 4D data. Traditional approaches primarily rely on a numerical scheme to solve the pressure-Poisson equation to obtain a dense pressure inference. This involves heavy expert intervention at each stage and requires significant computational resources. Concurrently, the application of current machine learning algorithms for solving partial differential equations is limited to domains with simple boundary conditions. We address these challenges in this paper and present V2P-Net: a novel, neural-network-based approach as an alternative method for inferring pressure from the observed velocity fields. We design an end-to-end hybrid-network architecture motivated by the conventional Navier-Stokes solver, which encapsulates the complex boundary conditions. It achieves accurate pressure estimation compared to the reference numerical solver for simulated flow data in multiple complex geometries of human in-vivo vessels.

1 Introduction

Imaging modalities such as 3D phase-contrast magnetic resonance imaging [14] or particle imaging velocimetry [23] enable us to get an in-vivo measurement of blood flow velocity. Relative pressure fields, inferred from the measured velocity fields, serve as a clinical biomarker for various cardiovascular diseases such as aortic valve stenosis, aortic coarctation, and aortic aneurysm. Additionally, the spatio-temporal distribution of pressure fields within a vessel segment is often used for clinical intervention and therapeutic planning in neurovascular diseases, such as cerebral angioma and intracranial aneurysm.

Given the velocity measurements, solving pressure fields simplifies the Navier-Stokes equation to the Pressure-Poisson Equation (PPE). [22] used an iterative

A. Feragen et al. (Eds.): IPMI 2021, LNCS 12729, pp. 545–558, 2021.
https://doi.org/10.1007/978-3-030-78191-0_42

scheme to solve the PPE. Heuristics-based attempts to compute a robust line integral have been proposed by [18]. [4] proposed an alternate method to the conventional PPE solution by using the work-energy conservation principle. [10] introduced a finite-element method (FEM) to solve the PPE and [15] improved upon this by incorporating flow-aware boundary tagging. A detailed analysis of the pressure estimation methods, in particular has been presented in [1].

In medical imaging, although mesh-based solutions are popular, they are computationally intensive. Recent machine learning-based solutions [9] are also compute-costly. Moreover, these methods demand expert intervention to accurately take care of patient-specific simulation domain and spatiotemporal discretization. Moving towards a fundamentally different direction from the prior literature, the goal of this paper is to propose an alternate, fast, and accurate pressure inference scheme without requiring the computationally expensive mesh-based analysis. In this paper, we mainly focus on neural network (NN)-based methods which can provide a good approximation of costly numerical schemes, such as the Navier-Stokes, for Partial Differential Equations (PDE).

Differential Equations (DE) have been studied in physics and engineering for quite a few decades. However, a strong connection between the NN and the DE has only been established very recently. While numerical schemes for solving DE can be exploited to design efficient network architectures, enforcing physics in an NN could also accelerate the numerical solutions of DE [3]. Recent methods based on a fully supervised training regime have shown promising results to learn the unknown physics model from conditional latent variables [8] and using a fully convolutional neural network (FCN) [5,20]. However, these methods are only favorable to solve the forward simulation problem. On the other hand, explicit prior information about the underlying physics can help an NN to solve forward and inverse problems in the PDE system. Two main categories of methods have emerged in recent times to incorporate physics in the network:

1. **Physics constraints in the architecture of the NN:** Classical approaches to solving a PDE system can be substituted by an approximation model that can be learned from the observed data. This formulation relies on a specific spatio-temporal discretization of a continuous-time process that offers the ability to generalize well over a variety of domains. [12] proposed one such generic network architecture (PDE-Net), which is parameterized by a learnable coefficient and constrained convolutional kernel to discover the underlying PDE from observed data. [21] has shown that an FCN can approximate well an intermediate step of the Euler velocity update rule for the inviscous Navier-Stokes equation by solving the Poisson equation.
2. **Physics constraints in the loss of the NN:** An alternate approach to obeying a PDE equation is to include it as a standalone loss or as an additional regularizer along with a data-fidelity loss. [16] show that prior knowledge about the form of a PDE in a loss function can help in exploiting differentiable programming (deep learning), to simulate a PDE and use the learned system to infer unknown system parameters. While this helps to obtain a robust fitting of a parameterized continuous-time generative model, the inclusion

of PDE system parameters in a domain-specific boundary loss restricts its generalization. Note that we refer to the coefficients of the PDE equation as the system parameters to differentiate it from the learnable NN parameters.

[21]'s method achieves significant acceleration for inviscid fluid flow simulation by using a semi-implicit Lagrangian convection scheme coupled with an FCN to approximate the traditional Jacobi iteration-based Poisson's equation solver. While this method shares our goal to infer pressure fields, they do not account for the change in velocity at the boundary (inflow-outflow) and the pressure drop from the viscous energy loss. Their proposed FCN assumes zero Neumann BC whereas, on the contrary, this is non-zero at the inlet and outlet boundaries of the flow as shown by [6]. Hence, it is not readily applicable for inferring pressure in the case of transient viscous fluid flow such as blood.

Our Contribution: In this paper, we present a proof-of-concept study of **V2P-Net** - a novel NN-based framework for inferring pressure from time-varying 3D velocity fields (Fig. 1). The two key properties of V2P-Net are: 1) inclusion of diffusion in an end-to-end trainable network that emulates the Navier-Stokes equation solver, and 2) decoupling the relative pressure into: a) convection induced pressure p_G, and b) inlet-outlet boundary driven pressure p_B. We show that p_G and p_B can be estimated using two separate neural networks dedicated to solving the Poisson's and Laplace's equations (Eqs. 3 & 4), respectively. p_G is modeled by solving the zero Neumann BC using a FCN, while a fully-connected NN (MLP) is used to map a geometry specific p_B conditioned on the non-zero Neumann BC. The FCN leverages the physics constraint in the network architecture, which enables it to generalize well over a different domain, while the MLP learns a domain-specific function by satisfying the non-zero Neumann BC as the loss function. Both of them are optimized by exploiting the underlying physics of the Navier-Stokes equation in an unsupervised way, thus obviating any need for ground-truth training data of the pressure distribution. We demonstrate the efficiency of the V2P-Net architecture on simulated cerebral blood flow geometries.

2 Methodology

2.1 Background

The Navier-Stokes equation for incompressible fluid in 3D volume is described by the following momentum-balance equation:

$$\underbrace{\frac{\partial \mathbf{u}}{\partial t}}_{\text{Transient}} + \underbrace{\nabla \mathbf{u} \cdot \mathbf{u}}_{\text{Convection}} = \underbrace{\nu \Delta \mathbf{u}}_{\text{Diffusion}} - \frac{1}{\rho}\nabla p + \mathbf{f};\ \text{subject to } \nabla \cdot \mathbf{u} = 0 \tag{1}$$

In a bounded domain Ω with Dirichlet BC, $\mathbf{u} = \mathbf{u}_b(\mathbf{x}, t)$ on boundary $\delta\Omega$; subject to $\int_{\delta\Omega} \boldsymbol{\eta} \cdot \mathbf{u}_b = 0$, ∇ and Δ represent the gradient and Laplacian operations

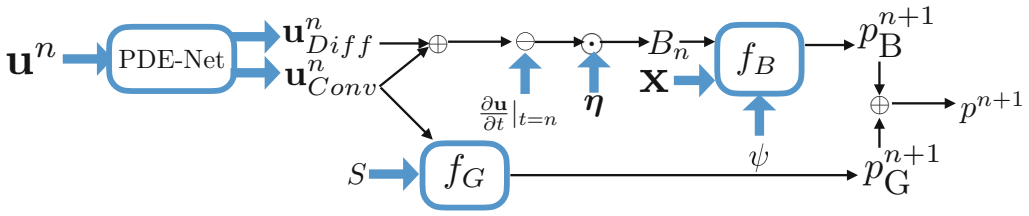

Fig. 1. Overview of V2P-Net. PDE-Net, f_G and f_B represent constrained convolutions, an FCN and an MLP respectively. Individual architecture of PDE-Net, f_G, and f_B are defined in detail in Fig. 2. S and B_n denote the vessel segmentation and BC respectively. $\mathbf{u}^n_{Conv}$ has a non-zero divergence field which is subsequently corrected using the convective induced pressure p^{n+1}_G.

respectively and $\boldsymbol{\eta}$ is the normal on $\delta\Omega$. In Eq. 1, $\mathbf{u}, p, \nu, \rho$ and, $\mathbf{f}$ denote velocity, pressure, kinematic viscosity, density of the fluid and external force respectively.

In our case, velocity ($\mathbf{u}$) is known and the pressure (p) is unknown. For a divergence-free flow, taking divergence of 1, we have to solve

$$\Delta p^{n+1} = \nabla \cdot (-\nabla \mathbf{u}^n \cdot \mathbf{u}^n), \text{ subject to } \frac{\partial p^{n+1}}{\partial \eta} = B_n \tag{2}$$

where the boundary condition (BC) $B_n = \boldsymbol{\eta} \cdot (\nu \Delta \mathbf{u}^n - \nabla \mathbf{u}^n \cdot \mathbf{u}^n - \frac{\partial \mathbf{u}}{\partial t}|_{t=n})$. In the following, we show that this equation can be modelled in a feed-forward NN, the V2P-Net (c.f. Fig. 1).

2.2 V2P-Net:

Here, we formulate the pressure estimation problem (Eq. 2) in the context of PDE approximation using an NN and describe our proposed *hybrid* NN architecture, V2P-Net, which consists of: 1) constrained convolution operations as **PDE-Net** [12], 2) an **FCN**, and 3) an **MLP** module. Parallels can be drawn between the proposed network and a recurrent neural network (RNN), where each time step update of the Navier-Stokes equation can be viewed as an unrolled step of the recurrence scheme. However, a conventional RNN would need explicit supervision for training, and therefore we use our hybrid network, which is unsupervised.

Convection-Diffusion Modelling (PDE-Net): The convection-diffusion $(\nabla \mathbf{u}^n \cdot \mathbf{u}^n, \Delta \mathbf{u}^n)$ operation is commonly solved in semi-implicit form for better numerical stability. However, semi-Lagrangian semi-implicit schemes lose the end-to-end trainability over multiple time-steps [21]. Therefore, we opt for an explicit convection-diffusion operation modeled as a *constrained convolution kernel* [12]. The constrained convolution represents a particular differential operator (in this case, convection and diffusion) as a learnable parameter in the explicit scheme. Thus, the network becomes end-to-end trainable even in the presence of viscosity over multiple time steps. We denote the resultant convection-diffusion

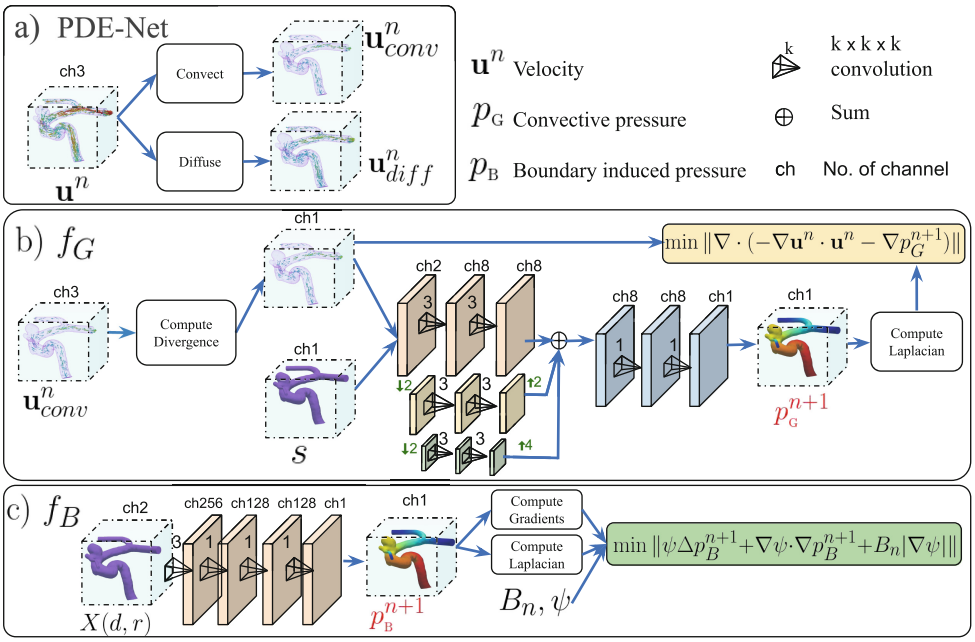

Fig. 2. Building blocks of our proposed method: a) The explicit convection-diffusion step is modeled using constrained convolution shown as *PDE-Net*, which enables an end-to-end differentiability. The resultant convective term ($\mathbf{u}^n_{conv}$) acts as a source for f_G, while the diffused term ($\mathbf{u}^n_{diff}$) contributes to the boundary condition B_n. b) f_G is a *FCN*, with ReLU activation, which takes the divergence of the convective velocity and segmentation mask S as input and produces the convective pressure, p_G. c) f_B is an *MLP* on channel dimension with *tanh* activation, which takes a domain descriptor $\boldsymbol{X}$ (Fig. 5) as input and maps it to a spatial function p_B. f_B is optimized for a specific geometry with their corresponding phase-field function ψ and specific boundary condition B_n.

operation ($\nabla\mathbf{u}^n \cdot \mathbf{u}^n$, $\Delta\mathbf{u}^n$) as ($\mathbf{u}^n_{Diff}$, $\mathbf{u}^n_{Conv}$). This part of the network is detailed as PDE-Net in Fig. 2a.

Pressure Decoupling (p_G, p_B): The resultant diffused and convective velocity from the PDE-Net ($\mathbf{u}^n_{Diff}$ and $\mathbf{u}^n_{Conv}$) in Fig. 2a contributes to the source of the pressure gradient. We make a few observations here: 1) $\nabla \cdot (-\nabla\mathbf{u}^n \cdot \mathbf{u}^n)$ causes a pressure gradient due to the geometric variation within the control volume. Additionally, the BC is non-zero at the inlet-outlet boundary and zero elsewhere. Thus, the pressure drop due to BC is global in nature for a particular computational domain. 2) One may question as to why we do not use an end-to-end FCN to infer pressure from velocity directly? To infer pressure from velocity, the computational domain plays an important role through the BC (Dirichlet for velocity and Neumann for pressure generally). Although FCNs are good for learning generalizable feature aggregates (from low-level edges to high-level objects in a hierarchical manner), we need to solve a volumetric non-local operation such as a boundary-conditioned volume integral to solve the pure Neumann BC. This requires the whole domain information at once, which is nontrivial for an FCN

to learn with a finite receptive field. 3) Similarly, we do not design an MLP-only model with the Navier-Stokes equation embedded in the loss [17] because, if the contribution in the pressure from the local geometric variation and global BC are not balanced, the MLP often overlooks the weaker contribution.

To overcome the above problems, we propose to decouple the total pressure, p^{n+1}, as a sum of two components, i.e. $p^{n+1} = p_G^{n+1} + p_B^{n+1}$ (Fig. 1), where p_G^{n+1} denotes the pressure quantity induced by the convection due to variation in local geometric shape and p_B^{n+1} represents the pressure distribution coming from the inflow-outflow BC. Henceforth, the PPE needs to solve two separate equations,

$$\text{Poisson's Equation: } \Delta p_G^{n+1} = \nabla \cdot (-\nabla \mathbf{u}^n \cdot \mathbf{u}^n); \text{s.t. } \frac{\partial p_G^{n+1}}{\partial \eta} = 0 \tag{3}$$

$$\text{Laplace's Equation: } \Delta p_B^{n+1} = 0; \text{ s.t. } \frac{\partial p_B^{n+1}}{\partial \eta} = B_n \tag{4}$$

For complex blood vessel geometries, the handling of BC is difficult due to: a) discretization error, b) error in the boundary estimate, and c) implementation of finite difference schemes for non-trivial BC, i.e., non-zero Neumann. [11] have shown a practical approximation for complex geometry by using a phase-field function for the Neumann BC in Eq. 4.

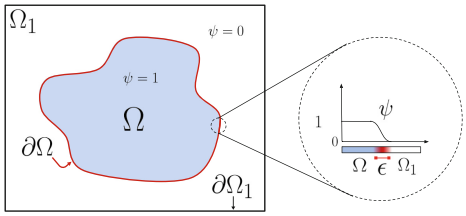

Fig. 3. An illustration of the phase-field function on a complex boundary. The original BC was Neumann at the boundary $\partial\Omega$. Diffused domain approximation transforms it into a Dirichlet BC at the modified simpler boundary $\partial\Omega_1$. The smoothness factor (ϵ) controls the approximation error vs robustness to noisy boundary trade-off.

As shown in Fig. 3, the computational domain can be realized as a phase-field function ψ, where $\psi = 1$ inside the domain and $\psi = 0$ outside the domain. At the boundary, this creates a smooth transition from $1 \rightarrow 0$ and therefore improves the handling of complex geometries.

We solve the following approximated PDE problem,

$$\psi \Delta p_B^{n+1} + \nabla \psi \cdot \nabla p_B^{n+1} + B_n |\nabla \psi| = 0 \tag{5}$$

This formulation transforms the Neumann BC into a source term in the modified advection-diffusion equation (Eq. 5) with spatially varying coefficients. Additionally, this alleviates the Lagrange multiplier tuning between the energy of the Laplacian of the predicted field inside the domain and the energy mismatch at the boundary as used in [13].

FCN & MLP Share the Burden: To predict the convective induced pressure p_G^{n+1}, we need to learn a relationship between the divergence source term to a compensating scalar field. Since the divergence occurs locally in nature, an FCN is an ideal candidate for this task. We incorporate the multi-resolution FCN proposed by [21] to solve Eq. 3. Let's denote this network as f_G parameterized by θ (as described in Fig. 2b). The predicted pressure, p_G^{n+1}, is given by,

$$p_G^{n+1} = f_G\left(\nabla \cdot \left(-\nabla \mathbf{u}^n \cdot \mathbf{u}^n\right), S; \theta\right) \tag{6}$$

where S is the binary segmentation mask of the vessel. We train f_G by minimizing the cumulative divergence of the predicted velocities over T time steps,

$$\mathcal{L}_{\mathcal{G}} = \frac{1}{T}\sum_{i=1}^{T}\int_{\Omega}\left\|\nabla \cdot \left(-\nabla \mathbf{u}^n \cdot \mathbf{u}^n \; - \frac{\delta t}{\rho}\nabla p_G^{i+1}\right)\right\| dv \tag{7}$$

As shown earlier, Eq. 4 ensures a smooth BC even for complex geometries. We, therefore, leverage the universal approximator property of the MLP to parameterize the functional form of the solution of Eq. 4 by f_B with a set of parameters ϕ. As shown in the MLP module in Fig. 2.c, f_b takes a description of the domain (Fig. 5) as input and maps it to a continuous spatial functional field. While in a simpler geometry, the Cartesian coordinate system is a good domain descriptor, a manifold aware descriptor is essential for the f_B to learn unctions, which are specific to geometries, such as blood vessels. To encode the manifold information in the domain discriminator, we employ a local cylindrical coordinate system as shown in Fig. 5. The f_B network architecture is explained in Fig. 2.c. Now, p_B^{n+1} is computed as follows

$$p_B^{n+1} = f_B(\mathbf{x}, B_n, \psi; \phi) \tag{8}$$

We adopt the phase-field approximation as described in the previous section to easily handle complex geometry and increase robustness against noisy BC. We train this network by minimizing the Laplacian energy over the control volume while obeying the BC.

$$\mathcal{L}_{\mathcal{B}} = \underbrace{\int_{\Omega} \|\psi \Delta p_B^{n+1} + \nabla\psi \cdot \nabla p_B^{n+1} + B_n|\nabla\psi|\| dv}_{\text{volume integral}} \tag{9}$$

Appropriate discretization is employed to numerically evaluate the volume and surface integral in Eq. 9. Note that both Eqs. 7 and 9 are trained in an unsupervised manner. Initial conditions, the BC and segmented vessel mask are sufficient inputs for training.

3 Experiments

In the following, we introduce a proof-of-concept study for validating our proposed method on a simulated blood flow for brain aneurysms. We aim to estimate the pressure within a blood vessel from the velocity fields for a viscous

and transient fluid such as blood. As we do not have the ground truth pressure to compare within the case of in-vivo 3D phase-contrast MRI, we rely on patient-specific simulated flow models to validate our method. Furthermore, due to the unavailability of standard datasets, it is a common practice [7,16,19] to validate the learning-based method by comparing it to a numerical reference solution. Therefore, given these constraints, we adopt a similar approach in our work to validate our proposed pressure estimation model against the solution of a numerical solver. Moreover, the unavailability of open-source codes posed a critical bottleneck for us to benchmark against competing methods.

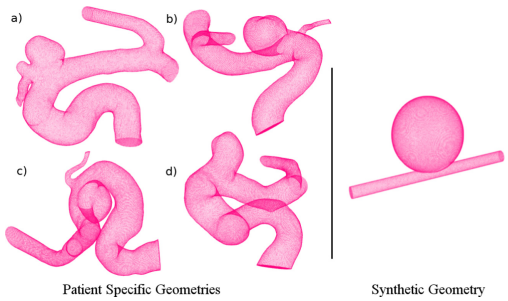

Fig. 4. (**Left**) Four patient-specific geometries ($a \rightarrow d$) are used for simulation of the flow segmented from 3D rotational angiogram images. These are the internal carotid arteries with aneurysms, which makes the data representative of complex blood vessel structures. After voxelization, the average size is $85 \times 110 \times 106$ voxels. (**Right**) Synthetic aneurysm geometry used for simulation of the flow. The spherical structure of the aneurysm induces vortex and complex flow which is used to train f_G

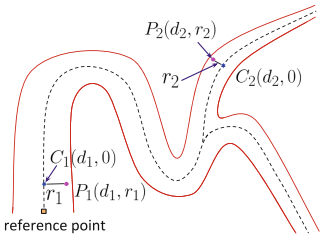

Fig. 5. Local cylindrical co-ordinates capture better domain topology than global Cartesian co-ordinates in case of blood vessel geometry as illustrated in the figure above. Points P_1 and P_2 are far apart to each other in the computational domain than the Euclidean distance between them. An efficient alternative way to describe the relative position is with respect to its distance from the nearest center-line points. Thus P_1 or P_2 can be represented with a pair of a) the distance from its closest centerline point, and b) the distance between its closest centerline point and the reference point along the centerline.

Flow Simulation: We use four patient-specific geometries with cerebral aneurysm (Fig. 4) to simulate 3D PC MRI sequences for one cardiac cycle, each with 24 time-points with temporal resolution of 36 ms and spatial resolution of 0.2 mm. We also have one synthetic geometry emulating an aneurysm in a uniform cylindrical vessel. We obtain two simulated flow sequences using the synthetic geometry, each with 100-time points and with a time step of 16 ms. The blood flow for each geometry was modeled as an unsteady Newtonian incompressible fluid with the following parameters: viscosity = 0.0032 Pa·s; density = 1050 kg/m^3. The Navier-Stokes equations were solved using the finite volume-based open-source platform OpenFOAM.V3.0. A second-order upwind scheme for the convective terms and a semi-implicit method for the pressure linked equations is used. The inlet patient-specific flow BC was extracted from 2D PC-MRI, and a zero pressure condition was used at the outlet. All the vascular walls were assumed rigid.

Training and Inference: Although the ground truth pressure is available at our disposal, we use them only for evaluation purposes. f_G is trained for direct inference of p_G (while ignoring p_B as it has no contribution in the inference of p_G due to the decoupling). We select a point for each geometry to use it as a reference point for quantitative and qualitative comparison. We only use two simulated sequences from the synthetic geometry for training f_G. Since f_B is a parameterized function conditioned by the specific BC, it is optimized on-the-fly during inference. Experimentally, the optimization of f_B converges after 40 iterations.

4 Results and Discussion:

Individual Performance of f_G and f_B (Ablation Study): For quantitative evaluation, we compare the predicted pressure, $p_G + p_B$, with the reference pressure for all four test geometries. Furthermore, we find that the contribution of p_B is more than that of p_G which is an expected behavior for highly viscous flow, such as blood, under transient acceleration. The total pressure is the only reference that is available from the CFD simulation and, therefore, we resort to additionally solving f_G using the Jacobi iteration and compare it with the prediction of f_G. This also serves as the ablation study of our approach since it evaluates f_G and f_B separately. Figure 8 shows box plots for the absolute error in total pressure ($p_G + p_B$) and p_G with respect to the CFD simulated reference pressure and reference p_G obtained from Jacobi method respectively. We observe that the highest error occurs at the most complex structure of geometry d.

From Fig. 6, we observe that the pressure estimation at the aneurysm boundary has a higher absolute error than the areas with high flow. The outliers observed in the Bland-Altman plot (Fig. 9) originate from this region. We hypothesize that the local-cylindrical coordinate is optimal for a healthy vessel-like structure but sub-optimal for degenerated cases as in aneurysm (which does not have any active inflow or outflow rather than turbulent vortex). Our future

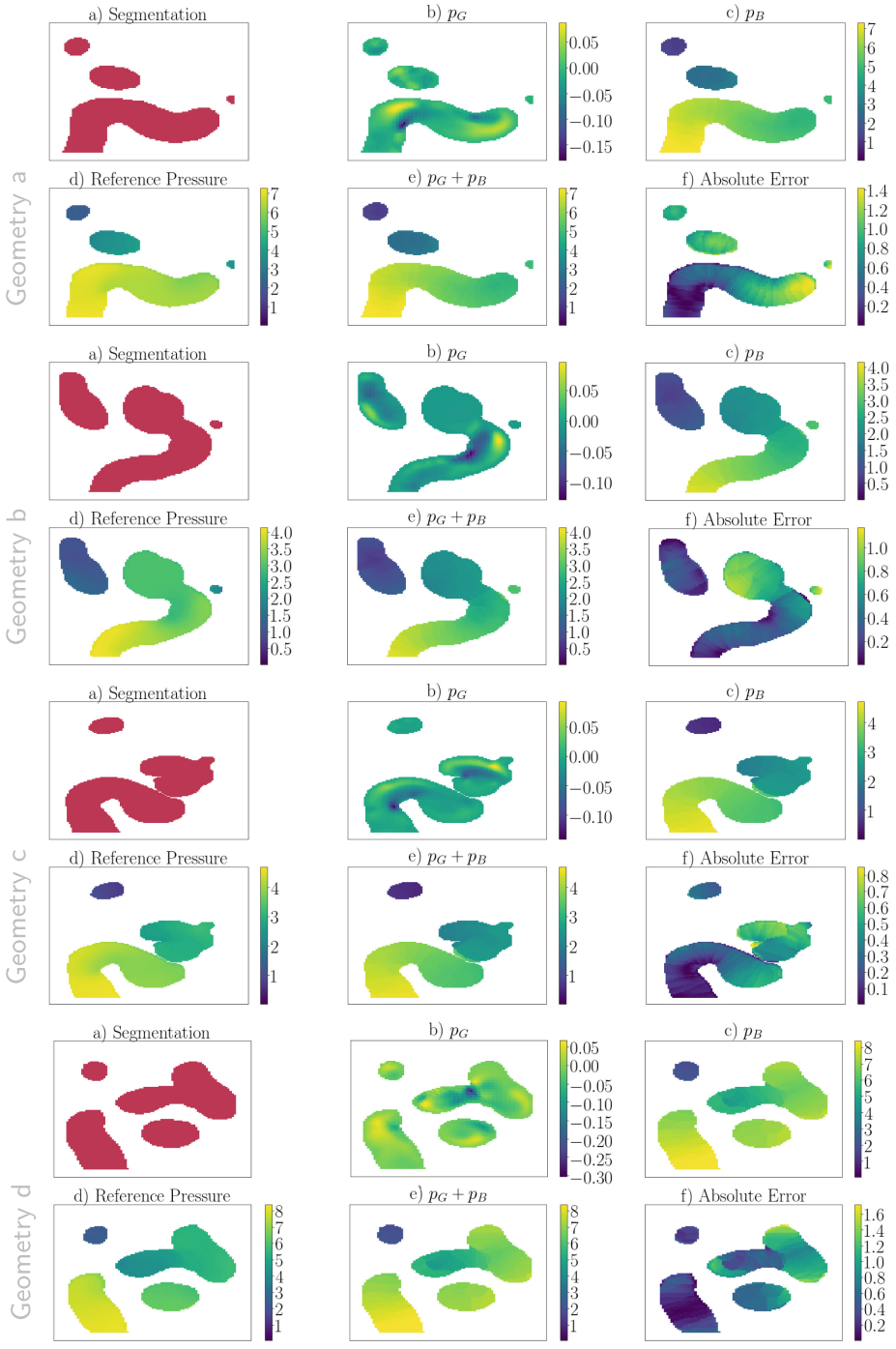

Fig. 6. An example qualitative comparison from the test sequence at a time instant (at t = 288 ms) from our experiment for all four geometries. All views are a 2D slice taken from the mid-section of the geometry. The unit for pressure is in mm Hg and the reference pressure at the primary outlet is set zero for the ground-truth, p_G, and p_B. a) shows the segmented mask. b) shows that f_G accurately captures the convective local pressure variation and c) shows that f_B infers a rather global pressure distribution induced by BC. d) shows simulated reference pressure. f) shows the absolute difference between $p_G + p_B$ (shown in e) and the reference pressure.

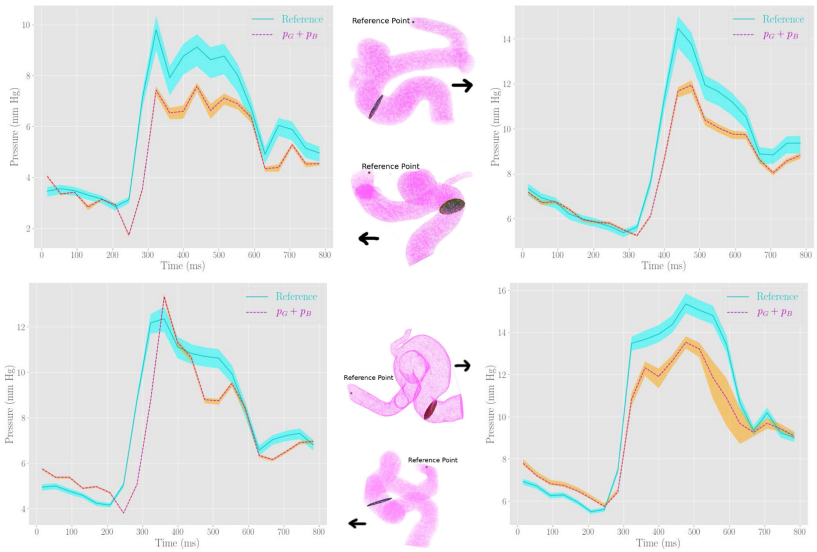

Fig. 7. Comparison between the mean reference and the mean estimated pressure for four different geometries at the slice locations shown in the middle. The banded curves indicate the 25% and 75% percentile of the distribution. The predicted pressure shows a good agreement in dynamic behaviour as the simulated reference.

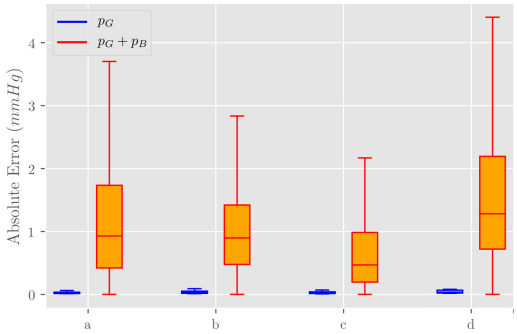

Fig. 8. Box plot for the absolute error in estimate of p_G and $p_G + p_B$ for four different geometries over one cardiac cycle. The median error for $p_G + p_B$ is consistently less than 1.5 mm Hg for all four geometries. Since, the contribution from p_B is higher compared to p_G, the error in $p_G + p_B$ mainly comes from p_B.

work will focus on learning a more general domain descriptor using geometric deep learning [2] suitable for a larger class of geometries than using hard-coded coordinates. This will facilitate the extension of our approach to arbitrary structures in multiple applications.

Distribution of Slice Dynamics: We examine the dynamic behavior of the predicted pressure at a particular plane over one cardiac cycle. The comparison depicted in Fig. 7 shows a good agreement between the simulated reference and the estimated pressure in terms of mean value and the distribution at the selected slice. We observe that our proposed method underestimates when the reference

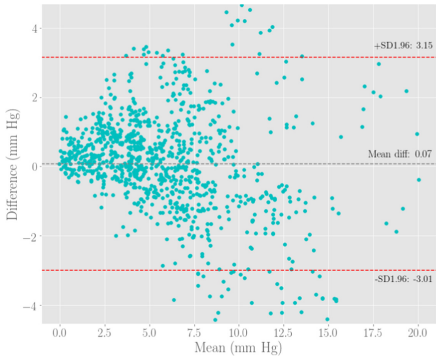

Fig. 9. Bland-Altman plot between 500 randomly chosen data points of the reference pressure and $p_G + p_B$ from all four geometry. The X-Axis represents the average of the reference pressure and $p_G + p_B$ estimates while the Y-axis represents their difference. This plot shows a high correlation between the estimated $p_G + p_B$ and the reference pressure with very few outliers.

pressure is high. We attribute this to be a performance trade-off as a consequence of the increased robustness offered by the phase-field approximation against the uncertainty in the BC values. The slice-wise processing is done as a post-processing step after the prediction using ParaView v5.6.0 software.

Run-Time Comparison: The mesh-based CFD simulation to generate reference data in a high performance computing cluster takes circa **10 h** for one subject per one cardiac cycle while the total inference time for out proposed method is circa **20 min** on a single Quadro P6000 GPU. A recent study based on an MLP where physics constraints work only in the loss function [9] has reported several hours (~ 7) to process a single geometry on a Tesla P100 GPU on the geometry of an aorta. Since they learn the convective source and the BC for every time points altogether, it takes a longer time for the network to learn the spatial distribution of pressure. On the contrary, our method leverages time discretization through network architecture and the splitting of pressure facilitates fast inference of p_G and spends most of its computing time only to infer p_B.

5 Conclusion

We introduce an end-to-end neural network approximating the traditional Navier-Stokes solver for incompressible Newtonian fluid flow with non-zero Neumann boundary conditions. Furthermore, in the context of pressure inference from the time-varying velocity field, we propose a novel approach for pressure decoupling and validate this using a generalizable FCN and an anatomy specific MLP-regressor based on the convection and boundary condition of the fluid. We evaluate the proposed method on simulated patient-specific blood flow data and find that our estimation closely approximates the simulated ground truth.

Acknowledgment. S. Shit and I. Ezhov are supported by the TRABIT Network under the EU Marie Curie Grant ID: 765148. D. Das is supported by the National Institute of Health (NIH) through the Nobrainer Project under the grant number: 1RF1MH121885-01A1. We thank NVIDIA for granting a Quadro P6000.

References

1. Bertoglio, C., et al.: Relative pressure estimation from velocity measurements in blood flows: SOTA and new approaches. Int. J. Numer. Meth. Bio. 34(2) (2018)
2. Bronstein, M.M., et al.: Geometric deep learning: going beyond euclidean data. IEEE Signal Process. Mag. **34**(4), 18–42 (2017)
3. Chen, T.Q., et al.: Neural ordinary differential equations. In: Proceedings of the NeurIPS. pp. 6571–6583 (2018)
4. Donati, F., et al.: Non-invasive pressure difference estimation from PC-MRI using the work-energy equation. Med. Image Anal. **26**(1), 159–172 (2015)
5. Ezhov, I., et al.: Geometry-aware neural solver for fast bayesian calibration of brain tumor models. arXiv preprint arXiv:2009.04240 (2020)
6. Gresho, P.M., Sani, R.L.: On pressure boundary conditions for the incompressible Navier-Stokes equations. Int. J. Numer. Methods Fluids. **7**(10), 1111–1145 (1987)
7. Hsieh, J.T., et al.: Learning neural PDE solvers with convergence guarantees. In: Proceedings of the ICLR (2019)
8. Kim, B., et al.: Deep fluids: a generative network for parameterized fluid simulations. In: Computer Graphics Forum, vol. 38, pp. 59–70. Wiley Online Library (2019)
9. Kissas, G., et al.: Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4d flow MRI data using physics-informed neural networks. Comput. Methods. Appl. Mech. Eng. **358**, (2020)
10. Krittian, S.B., et al.: A finite-element approach to the direct computation of relative cardiovascular pressure from time-resolved MR velocity data. Med. Image Anal. **16**(5), 1029–1037 (2012)
11. Li, X., et al.: Solving PDEs in complex geometries: a diffuse domain approach. Commun. Math. Sci. **7**(1), 81 (2009)
12. Long, Z., et al.: PDE-net: learning PDEs from data. In: Proceedings of the 35th ICML, vol. 80, pp. 3208–3216. PMLR (2018)
13. Magill, M., et al.: Neural networks trained to solve differential equations learn general representations. In: Proceedings of the NeurIPS, pp. 4071–4081 (2018)
14. Markl, M., et al.: 4D flow MRI. JMRI **36**(5), 1015–1036 (2012)
15. Mihalef, V., et al.: Model-based estimation of 4D relative pressure map from 4D flow MR images. In: STACOM. pp. 236–243. Springer (2013)
16. Raissi, M., et al.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
17. Raissi, M., et al.: Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. Science (2020)
18. Rengier, F., et al.: Noninvasive pressure difference mapping derived from 4D flow MRI in patients with unrepaired and repaired aortic coarctation. Cardiovascular Diagnosis Therapy **4**(2), 97 (2014)
19. Shit, S., et al.: Implicit neural solver for time-dependent linear pdes with convergence guarantee. arXiv preprint arXiv:1910.03452 (2019)

20. Thuerey, N., et al.: Deep learning methods for reynolds-averaged navier-stokes simulations of airfoil flows. AIAA Journal pp. 1–12 (2019)
21. Tompson, J., et al.: Accelerating Eulerian fluid simulation with convolutional networks. In: Proceedings of the 34th ICML, vol. 70, pp. 3424–3433. PMLR (2017)
22. Tyszka, J.M., et al.: Three-dimensional, time-resolved (4D) relative pressure mapping using magnetic resonance imaging. JMRI **12**(2), 321–329 (2000)
23. Van Oudheusden, B.: PIV-based pressure measurement. Measur. Sci. Technol. **24**(3), 1–32 (2013)

SRflow: Deep Learning Based Super Resolution of 4D-flow MRI Data

Suprosanna Shit, Judith Zimmermann, Ivan Ezhov, Johannes C. Paetzold, Augusto F. Sanches, Carolin Pirkl & Bjoern H. Menze

Journal: Frontiers in Artificial Intelligence

Synopsis: Exploiting 4D-flow MRI data for quantification of hemodynamics requires an adequate spatio-temporal vector field resolution at a low noise level. We provide a learned solution to super-resolve in vivo 4D-flow MRI data at a post-processing level to address this challenge. We propose a deep convolutional neural network that learns the inter-scale relationship of the velocity vector map and leverages an efficient residual learning scheme to make it computationally feasible. A novel, direction-sensitive, and robust loss function is crucial to learning vector-field data. We present a detailed comparative study between the proposed super-resolution and the conventional cubic B-spline based vector-field super-resolution. Our results demonstrate that peak-velocity to noise ratio, normalized root mean square error of speed, direction error, and root mean square divergence of the flow field improves significantly over the state-of-the-art cubic B-spline with $10x$ faster inference. More importantly, the proposed approach for super-resolution of 4D-flow data is capable of improving the reliable calculation of hemodynamic quantities.

Contributions of thesis author: Conceptualized the project, gathered necessary software resource, developed the novel loss function and implemented it, performed a leading role in the experiment, lead role in writing the manuscript.

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 12 August 2022
DOI 10.3389/frai.2022.928181

OPEN ACCESS

EDITED BY
Tien Anh Tran,
Viet Nam Maritime University, Vietnam

REVIEWED BY
Takaaki Sugino,
Tokyo Medical and Dental University, Japan
Julio Garcia,
University of Calgary, Canada

*CORRESPONDENCE
Suprosanna Shit
suprosanna.shit@tum.de

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Artificial Intelligence

RECEIVED 25 April 2022
ACCEPTED 11 July 2022
PUBLISHED 12 August 2022

CITATION
Shit S, Zimmermann J, Ezhov I, Paetzold JC, Sanches AF, Pirkl C and Menze BH (2022) SRflow: Deep learning based super-resolution of 4D-flow MRI data.
Front. Artif. Intell. 5:928181.
doi: 10.3389/frai.2022.928181

SRflow: Deep learning based super-resolution of 4D-flow MRI data

Suprosanna Shit[1,2]*, Judith Zimmermann[1], Ivan Ezhov[1], Johannes C. Paetzold[1], Augusto F. Sanches[3], Carolin Pirkl[1] and Bjoern H. Menze[2]

[1]Department of Informatics, Technical University of Munich, Munich, Germany, [2]Department of Quantitative Biomedicine, University of Zurich, Zurich, Switzerland, [3]Institute of Neuroradiology, University Hospital LMU Munich, Munich, Germany

Exploiting 4D-flow magnetic resonance imaging (MRI) data to quantify hemodynamics requires an adequate spatio-temporal vector field resolution at a low noise level. To address this challenge, we provide a learned solution to super-resolve *in vivo* 4D-flow MRI data at a post-processing level. We propose a deep convolutional neural network (CNN) that learns the inter-scale relationship of the velocity vector map and leverages an efficient residual learning scheme to make it computationally feasible. A novel, direction-sensitive, and robust loss function is crucial to learning vector-field data. We present a detailed comparative study between the proposed super-resolution and the conventional cubic B-spline based vector-field super-resolution. Our method improves the peak-velocity to noise ratio of the flow field by 10 and 30% for *in vivo* cardiovascular and cerebrovascular data, respectively, for 4× super-resolution over the state-of-the-art cubic B-spline. Significantly, our method offers 10x faster inference over the cubic B-spline. The proposed approach for super-resolution of 4D-flow data would potentially improve the subsequent calculation of hemodynamic quantities.

KEYWORDS

4D-flow MRI, residual learning, flow super-resolution, cerebrovascular flow, flow quantification

1. Introduction

Assessing quantitative hemodynamic metrics is crucial in diagnosing and managing flow-mediated vascular pathologies. For example, monitoring wall shear stress along the aortic vessel wall supports diagnostic assessment in patients with bicuspid aortic valves (Guzzardi et al., 2015; Garcia et al., 2019); alteration in pressure distribution (Leidenberger et al., 2020) is observed in Marfan disease. Similarly, local characterization of the vortex core pattern can assist in rapture risk estimation of the vascular aneurysm (Futami et al., 2019). 4D-flow magnetic resonance imaging (4D-flow MRI) (Markl et al., 2012) provides spatiotemporally resolved velocity vector maps of coherent blood flow through vascular structures. In applications mentioned above, 4D-flow MRI serves as a basis for quantifying flow parameters and patterns non-invasively.

Accurate computation of image-based quantitative hemodynamic metrics from 4D-flow MRI is limited by the trade-offs between spatiotemporal resolution, signal-to-noise ratio, and the clinically acceptable *in vivo* acquisition duration.

In particular, low spatial resolution at near-wall points hamper the numerical estimation of spatial derivatives of the three-dimensional vector field (Petersson et al., 2012). Furthermore, sometimes image registration is required to standardize the image space for comparison purposes (Cibis et al., 2017), which also involves isotropic resampling or super-resolution of the 4D-flow data. Moreover, clinically important qualitative visualization, such as streamlines (Cebral et al., 2011) and vortex core line (Byrne et al., 2014) delineation, relies on improved spatial resolution. Therefore, having access to high-resolution 4D-flow MRI image data is critical to infer hemodynamic metrics reliably and using computational routines for improving the image data after acquisition has become an indispensable step in processing these data. This brings to the generations of efficient acquisition algorithms and hardware acceleration, such as parallel imaging (Stankovic et al., 2014), non-Cartesian trajectories (Markl et al., 2012), k-t SENSE (Tsao et al., 2003), k-t GRAPPA (Breuer et al., 2005), which enabled 4D-flow MRI. In addition, several algorithms have emerged over time, building on this accelerated acquisition to enhance the measured 4D-flow MRI. In this direction, there are two parallel streams of research in the context of MRI super-resolution: 1) Compressed sensing: MRI super-resolution by improving high-frequency components from the k-space (Santelli et al., 2016; Ma et al., 2019) and 2) Single-volume MRI super-resolution: acquiring conventional MRI at low spatiotemporal resolution and retrospectively super-resolve data at the image level (Ferdian et al., 2020) as post-processing.

Previous works based on MRI image quality enhancement from k-space were focused toward velocity-field denoising (Ong et al., 2015), divergence reduction (Mura et al., 2016), intravoxel dephasing (Rutkowski et al., 2021), and streamline denoising (Callaghan and Grieve, 2017). In contrast, super-resolution as a post-processing step in the image space is more versatile and applicable to any collection of images acquired by differing sequences or MRI scanners, agnostic to the specifics of the k-space sampling. Once we have the reconstructed MRI images in the form of DICOM or NIFTI, super-resolution in image space is an efficient and hustle-free plug-and-play feature. As such, it is not a competing but a complementary and independent field of work that is of particular relevance when dealing with large and inhomogeneous multi-centric data sets or when access to original k-space recordings is not available. In spite of an abundance of conceptually related machine learning-based techniques for video, super-resolution (Chu et al., 2018) and optical-flow estimation (Liu et al., 2019), (that has not been used for 4D-flow MRI super-resolution, though) the data-driven reconstruction in image space remains under-explored in 4D-flow MRI, which is a commonly used method either rely on 4D cubic spline (Stalder et al., 2008; Dyverfeldt et al., 2014) or sinc (Bernstein et al., 2001) interpolation. In this work, we will focus on adapting deep learning-based super-resolution to the specific requirements of 4D-flow interpolation, leveraging prior knowledge of flow fields from prior observations. At the same time, we identify that the loss function is crucial for this translation and offers a novel loss well-adapted to velocity fields.

1.1. Prior work on super-resolution in image space

Super-resolution is a well-studied topic in computer vision, where high-resolution images are reconstructed from low-resolution images. Recently, deep neural networks based on super-resolution (Bhowmik et al., 2018) have become popular due to their high accuracy and fast processing time. Dong et al. (2016) first introduced a fully convolutional network for super-resolution. Most of the subsequent super-resolution approaches rely on residual learning, where we predict the fine detail using a convolutional network and add them with coarse upsampled images (i.e., cubic spline). Two distinct approaches in residual learning for super-resolution evolved in recent times: a) upsample in the beginning and then extract fine details from it using residual learning (Kim et al., 2016) and b) extract powerful image features from the low-resolution image and add them with the upsampled image at the end (Lim et al., 2017). The former enjoys extra performance improvement, while the latter is more efficient regarding the computational budget. Recently, channel widening before activation in the residual branch has been proposed (Yu et al., 2018), which not only helps shallow features to propagate easily into deeper layers but also reduces the network complexity. Zhang et al. (2018) has proposed a residual in residual architecture for a very deep network with a channel attention layer, which exploits non-linear interaction between global channel statistics to scale individual features.

1.2. Prior work on MRI super-resolution and challenges

Volumetric MRI super-resolution (Pham et al., 2017; Lyu et al., 2020) in image space is analogous to the 2D counterpart. However, the challenge lies in designing memory- and computation-efficient methods suitable for 3D volume that can be trained on a limited amount of training data. This problem is exaggerated for 3D vector-valued data, requiring new approaches to learn the inter-scale transformation effectively. Previously, in an attempt to mitigate the 3D computational complexity, a variation of residual learning called densely connected convolutional network has been adopted in MRI super-resolution by Chen et al. (2018). Often super-resolution is interpreted as a texture synthesis problem using adversarial learning (Sánchez and Vilaplana, 2018). Although adversarial learning produces perceptually high-quality images (Xie et al., 2018), it fails to achieve superior reconstruction metrics

compared to non-adversarial learning, which is of main interest in the case of 4D-flow MRI to accurately compute the velocities and their spatial derivatives. Hence, we are not considering adversarial learning. Previously, data-driven super-resolution approaches explored other MRI modalities, such as super-resolution of temporal perfusion MRI (Meurée et al., 2019) and vector field super-resolution problem of diffusion MRI (Tanno et al., 2017; Albay et al., 2018). Note that these methods either rely on 2D slice-wise super-resolution or individual channel-wise super-resolution in 3D. Ferdian et al. (2020) proposed a residual network to super resolve 4D Flow MRI. Although they also used computational fluid dynamic (CFD) to mitigate the shortcomings of noisy *in vivo* data, their method relies on the magnitude image alongside the velocity image. For CFD, it is hard to simulate a magnitude image due to the unknown relationship between the simulated velocity field and the magnitude image intensities. Fathi et al. (2020) proposed physics-informed deep learning to super resolve patient-specific flow. However, they need to retrain their model for each new patient since the physics-informed model does not generalize over the computation domain, i.e., the vessel geometry in this case.

1.3. Related work on 4D-flow MRI and CFD

Several classical approaches have been applied to improve the spatial resolution of 4D-fLow MRI, such as ridge-regression (Bakhshinejad et al., 2017) and Lasso regression (Fathi et al., 2018). Recently, Rutkowski et al. (2021) proposed a machine learning-based solution to merge the CFD and MRI data. Flow data assimilation is another active research field, where the reduced-order Kalman filter (Habibi et al., 2021) and local ensemble Kalman filter (Gaidzik et al., 2019, 2021) have been used. Incorporating CFD simulation using interior-point optimization framework (Töger et al., 2020) and lattice Boltzmann-based topology optimization (Klemens et al., 2020). While these works attempted to improve 4D-flow MRI by merging CFD data, it requires expensive CFD simulation for each new acquisition. Hence, we look for an alternative road where we can learn a model using both CFD and *in vivo* 4D-flow MRI data, and the learned solution can be used out of the box for any newly acquired data without any further CFD simulation.

Along this line, we aim to find an elegant solution to learn a scalable non-linear mapping from coarse- to fine-scale spatial velocity field. Further, three channels of 4D-flow MRI together represent the flow direction in 3D and should be treated as a joint interpolation problem compared to earlier approaches on scalar volumes (Chen et al., 2018; Sánchez and Vilaplana, 2018). Moreover, the commonly used ℓ_2 loss is sub-optimal for 4D-flow MRI because of the non-Gaussian (Gudbjartsson and Patz, 1995) noise distribution and does not prioritize the direction of point-wise velocity fields. Previously, direction-sensitive loss functions, such as cosine similarity, have been explored in text processing (Li and Han, 2013) and face recognition (Nguyen and Bai, 2010). Since in 4D-flow MRI, the flow direction consistency is important for all subsequent applications, we identify it as a crucial aspect and propose including it in a novel *mutually-projected* ℓ_1 loss function.

1.4. Our contribution

In summary, our contributions are as follows:

- We propose a novel and memory-efficient end-to-end convolutional neural network architecture, which learns the non-linear relationship between fine- and coarse-scale velocity fields and achieves super-resolution of the velocity field. Moreover, it applies to 4D-flow data irrespective of the scanner-specific constraints and access to the k-space information.
- We introduce a novel, robust, and direction-dependent cost function referred to as *mutually projected* ℓ_1. We investigate its effect on the proposed network compared to the standard ℓ_1 loss function.
- We further validate our method on *in vivo* 4D flow MRI datasets of two anatomical regions, namely: a) an internal carotid artery (ICA) brain aneurysm (Cerebrovascular data) and b) whole heart and great vessel (Cardiovascular data) that were acquired with different MRI scanners and at different imaging centers. This assesses the generalizability of the proposed method.

2. Materials and methods

In this section, we describe in detail the proposed learning-based method (Section 2.1). Subsequently, we describe the proposed robust loss function (Section 2.2) along with its implementation details (Section 2.3).

2.1. Network architecture

4D-flow MRI provides time-resolved 3D blood flow velocity maps over a single cardiac cycle. Our work focuses on super-resolving along the spatial dimensions and treats each temporal image frame as an independent sample. Let us denote the low resolution velocity field and high resolution velocity field as $\mathbf{u}$ and $\mathbf{U}$ respectively, where $\mathbf{u} \in \mathcal{R}^{H\times W\times D\times 3}$, $\mathbf{U} \in \mathcal{R}^{sH\times sW\times sD\times 3}$, H, W, D are the spatial dimensions, and s is the factor of upscaling ($s = 2, 3$ or 4). We are interested in learning a supervised data-driven mapping function from $\mathbf{u} \rightarrow \mathbf{U}$ from the input-output pairs $\{\mathbf{u}_i, \mathbf{U}_i\}_{i=1}^{n}$. Since three channels denote three velocity components are highly correlated, we

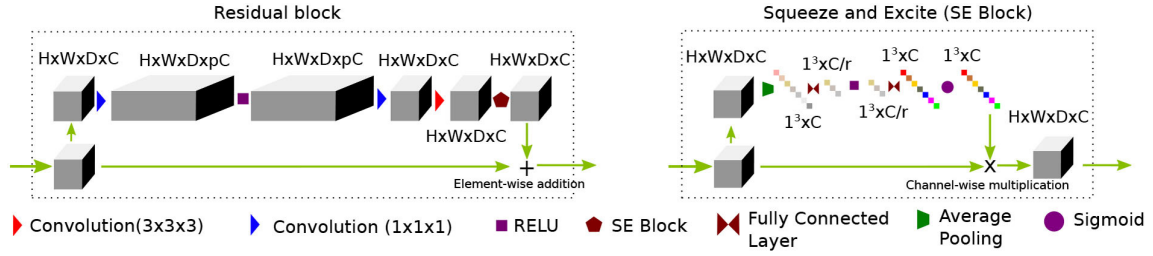

FIGURE 1
A proposed residual block consists of a sequence of channel widening, convolution followed by activation, channel squeeze and excitation (SE) block. While the channel widening by a factor of p before activation helps to reduce network parameters, the SE block promotes diverse feature distribution by compressing interim features by a factor of r.

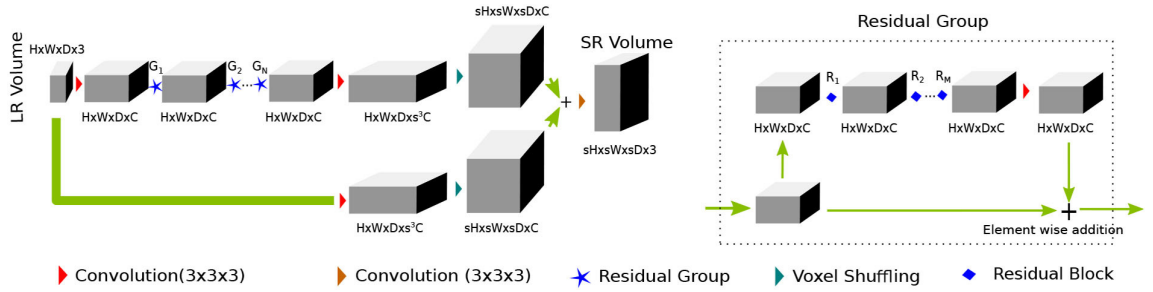

FIGURE 2
PA proposed SRflow has a series of residual groups made of residual blocks, as shown in Figure 1. The residual grouping strategy maximally leverages residual learning by exploiting hierarchical skip connections.

opted for a three-channel volumetric super-resolution instead of individually super-resolving each velocity component. To reduce the computational overhead, we naturally opt to extract a rich feature from the low-resolution vector field using residual learning and add the predicted fine details with upsampled vector fields to reconstruct high-resolution velocity fields.

2.1.1. Building blocks

In recent times (Yu et al., 2018), wide-activation before the convolution operation in the residual branch proved to be an efficient strategy to reduce model complexity without sacrificing super-resolution performance. We first extend the idea of having widely activated residual blocks in 3D. Henceforth, we will call this network WDSR-3D and use it as a baseline method for evaluation purposes. We keep the weight normalization as was proposed in Yu et al. (2018). We incorporate a sequence of residual groups, which effectively accelerates the learning of deep networks (Zhang et al., 2018). Since channel expansion with a $1\times1\times1$ convolution kernel is applied, new channels are different linear combinations of the previous layer channel without any spatial feature propagating in the channel dimension. Thus, information in the new channels carries redundant information. We argue that although this redundancy gives multiple paths for the gradient to propagate easily throughout the residual blocks, they often share similar information. The recently introduced squeeze and excitation (SE) (Hu et al., 2018) block helps to inject useful cross-channel diversity in the residual features. Hence, we leverage the feature diversity of an SE block to re-calibrate the channel features. This is depicted in Figure 1. We will refer to this modified architecture as SRflow in the subsequent discussion, which is depicted in Figure 2.

2.1.2. SRflow

The first convolution layer transforms the input to C channel feature maps. After first convolution, it goes through the N number of residual groups $\mathbf{G_1}, \mathbf{G_2}, \cdots, \mathbf{G_N}$. Each of the residual group consists of M number of $\mathbf{R_1}, \mathbf{R_2}, \cdots, \mathbf{R_M}$ residual blocks. The deep features and the input goes through their respective feature refining convolution layer, which transforms channels $C \rightarrow s^3C$. We use voxel shuffling layer [3D pixel shuffling (Shi et al., 2016) layer] to rearrange features from channel dimension to increase spatial dimension $sH \times sW \times sD \times C$. The final convolution layer merges fine details with the coarse up-scaled branch and reconstruct super-resolved volume of size $sH \times sW \times sD \times 3$.

10.3389/frai.2022.928181

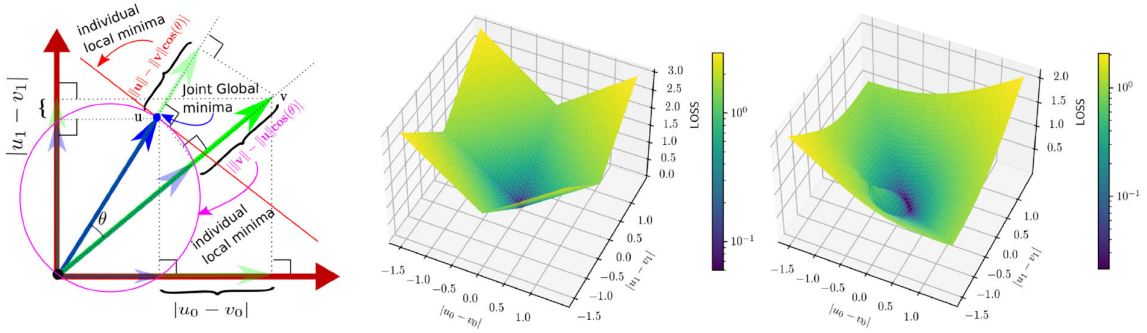

FIGURE 3
Left: A 2D delineation of *mutually projected* ℓ_1 (*mp*-ℓ_1) error. The reference and the predicted vectors are $\mathbf{u} = [u_0, u_1]$ and $\mathbf{v} = [v_0, v_1]$, respectively. Individual local minimas of $\mathcal{J}_1(\mathbf{v}; \mathbf{u})$ and $\mathcal{J}_2(\mathbf{v}; \mathbf{u})$ w.r.t $\mathbf{v}$ are shown in red line and purple circle, respectively. However the common minima of the two losses are unique and are as same as the reference. Middle: A description of the ℓ_1 error surface at $\mathbf{u} = [0.5, 0.5]$. Right: A description of the *mp* − ℓ_1 error surface for the same $\mathbf{u}$ and $\alpha = \beta = 0.5$, which has a direction sensitivity based on the value of $\mathbf{u}$.

2.2. Robust loss function

4D-flow MRI data acquired in clinical settings are sparse in both space and time, and they can be easily corrupted by noise. Since noise in 4D-flow MRI is not Gaussian (Gudbjartsson and Patz, 1995), it is sub-optimal to use an ℓ_2 norm on the error in our task. Thus, a robust cost function is needed for obtaining accurate estimates of the super-resolved flow. Let us denote the reference velocity as $\{\mathbf{u}_i\}_{i=1}^n | \mathbf{u}_i \in \mathcal{R}^3$ and the estimated velocity as $\{\mathbf{v}_i\}_{i=1}^n | \mathbf{v}_i \in \mathcal{R}^3$. The most commonly used robust loss function is ℓ_1 loss. For n number of samples, it is defined as

$$\mathcal{J}_{\ell_1} = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{v}_i - \mathbf{u}_i\|_1 \tag{1}$$

2.2.1. Mutually projected ℓ_1 loss

ℓ_1 loss penalizes the estimation error equally, irrespective of the reference vector direction. Neighboring voxels tend to have a different correlation in magnitude and direction based on local blood vessel geometry and the global flow direction. Because of this, we argue that a magnitude/direction disentanglement in the loss would benefit the network to arrive at a better trade-off between the accuracy of magnitude and direction estimation under noisy circumstances. Also, errors in magnitude and direction are very different in the value range across the spatial location, making it difficult to find optimal weight in the case of a weighted loss function. To overcome this, we propose to incorporate a directional sensitivity for the reference velocity in the loss function. Specifically, we introduce *mutually projected* ℓ_1 (*mp*-ℓ_1) error (c.f. Figure 3). The projected ℓ_1 error of $\mathbf{v}$ on $\mathbf{u}$ is given by

$$\mathcal{J}_1(\mathbf{v}; \mathbf{u}) = |\|\mathbf{u}\| - \|\mathbf{v}\| \cos(\theta)| \tag{2}$$

Where θ is the angle between $\mathbf{u}$ and $\mathbf{v}$. The local minima of $\mathcal{J}_1(\mathbf{v}; \mathbf{u})$ is the orthogonal subspace of $\mathbf{u}$. Similarly, the projected ℓ_1 error of $\mathbf{u}$ on $\mathbf{v}$ is given by

$$\mathcal{J}_2(\mathbf{v}; \mathbf{u}) = |\|\mathbf{v}\| - \|\mathbf{u}\| \cos(\theta)| \tag{3}$$

The local minima of $\mathcal{J}_2(\mathbf{v}; \mathbf{u})$ is the sphere centered at $\mathbf{u}/2$ with radius $\|\mathbf{u}\|/2$ excluding the origin $\mathbf{0}$. We take a convex linear combination of $\mathcal{J}_1$ and $\mathcal{J}_2$ to construct the $\mathcal{J}_{mp\text{-}\ell_1}$ loss

$$\mathcal{J}_{mp\text{-}\ell_1} = \frac{1}{n} \sum_{i=1}^{n} (\alpha \mathcal{J}_1(\mathbf{v}_i; \mathbf{u}_i) + \beta \mathcal{J}_2(\mathbf{v}_i; \mathbf{u}_i)) \tag{4}$$

where $[\alpha + \beta = 1 : 0 < \alpha, \beta < 1]$

Note that both $\mathcal{J}_1$ and $\mathcal{J}_2$ independently have two different solution spaces (cf Figure 3-left); however, $\mathcal{J}_{mp\text{-}\ell_1}$ has a unique local minima, which is also the global minima achieved under the condition of $\mathbf{v} = \mathbf{u}$. Figure 3 explains *mp*-ℓ_1 in 2D scenario. The same interpretation holds in a higher dimension with a hyper-sphere and a hyper-plane instead of a circle and a line. Unlike ℓ_1 loss, *mp*-ℓ_1 has directional sensitivity depending on the value of $\mathbf{u}$, which helps the pointwise error to adapt locally near the minima.

2.2.2. Combined loss

In our experiment, we find that the combination of ℓ_1 and *mp*-ℓ_1 losses helps to achieve the best performance in terms of training loss and validation PVNR. The complete loss function is as follows

$$\mathcal{J}_{opt} = \lambda_{\ell_1} \mathcal{J}_{\ell_1} + \lambda_{mp\text{-}\ell_1} \mathcal{J}_{mp\text{-}\ell_1} \tag{5}$$

where λ_{ℓ_1} and $\lambda_{mp\text{-}\ell_1}$ are two weight parameters.

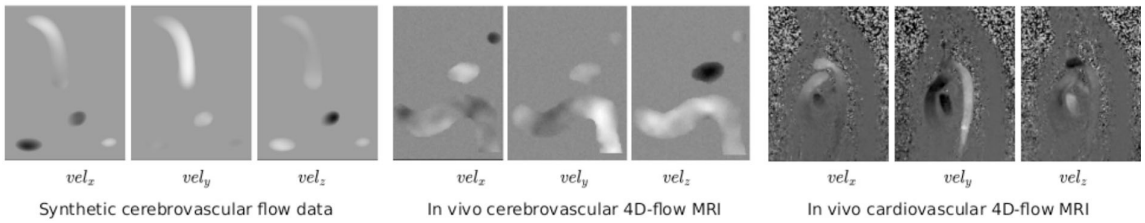

FIGURE 4
Typical examples of three data sets used in our experiments.

2.3. Implementation details

We implement our model in PyTorch. In the network architecture, we use $N = 4$ and $M = 2$. We expand the features by $p = 32$ times for the wide activation, and for the SE block, we use $r = 8$. For loss function, we select $\alpha = \beta = 0.5$ and $\lambda_{\ell_1} = \lambda_{mp\text{-}\ell_1} = 1$. We select the learning rate at 10^{-3} and use the ADAM optimizer for all of our experiments. We train each model for 200 epochs with a learning rate decay of 0.9 after every 10 epochs with a batch size of 4 in a Quadro P6000 GPU. The best model is chosen based on the validation peak velocity-to-noise ratio (PVNR).

2.4. Datasets

To study how well our model generalizes in practice, we chose two different vascular regions, which are also of numerous clinical relevance (Amili et al., 2018; Garcia et al., 2019). Importantly, these two datasets are obtained from two different scanners. The dataset consists of three different sets of flow data; a) **Synthetic Cerebrovascular Data:** CFD simulated flow data of cerebral aneurysm, b) ***In vivo* Cerebrovascular Data:** *in vivo* 4D-flow MRI of ICA aneurysm, and c) ***In vivo* Cardiovascular Data:** *in vivo* 4D-flow MRI of the whole heart and great vessels. Exemplary samples from the datasets have been shown in Figure 4.

2.4.1. Synthetic cerebrovascular data

We obtain patient-specific cerebrovascular aneurysms (N = 6) in ICA geometries from 3D rotational angiograms. We segment the blood vessel geometries from the computed tomography using the MITK v2018.4. We generate the triangulated mesh using ICEM CFD v.19 (ANSYS Inc). We model the blood flow as an unsteady Newtonian flow and solve the Navier-Stokes equations using the finite volume-based OpenFOAM-v3.0. We impose the inlet patient-specific flow boundary conditions extracted from 2D phase-contrast MRI and a zero pressure condition at the outlet. All the vascular walls are assumed rigid. We use a second-order upwind scheme for the convective terms and a semi-implicit method for pressure-linked equations. We employ an algebraic multi-grid-based solver for high-precision simulation. We simulate the blood flow with the following parameters: viscosity 0.0032 Pa·s; density 1,050 kg/m^3 (Brindise et al., 2019).

2.4.2. *In vivo* 4D-flow MRI data

A total number of 24 *in vivo* 4D-flow MRI data sets are included: Cardiovascular data of healthy subjects covering the whole heart (N = 10) or thoracic aorta only (N = 11); and cerebrovascular data of patients with ICA aneurysm (N = 3). All *in vivo* volunteers were recruited prospectively. The institutional review board approves all imaging studies, and written consent is obtained before scanning. All cardiovascular data are acquired using a 1.5 T MRI system (Siemens Avanto) with breathing navigator gating and prospective electrocardiogram triggering. Three ICA aneurysm data sets are acquired using a 3 T MRI system (Philips Achieva TX) with prospective cardiac triggering using a peripheral pulse unit. No contrast agent is used. Acquisition parameters for both types of data are listed in Table 1. For all datasets, Maxwell terms and gradient non-linearity are corrected during reconstruction. Eddy current phase offset is corrected offline.

2.5. Data preparation

For synthetic and *in vivo* data sets, we rely on the following data preparation steps to create training data. We consider the acquired image volume as the high-resolution reference data.

1. We first convert the velocity data into phase using a $venc = v_{max}/\pi$ to avoid any phase warping. Next, we combine the magnitude with phase and transform the reference data into Fourier space. Note that we use synthetic data's segmentation mask as the dummy magnitude.
2. Then, we crop the low-frequency component from the k-space according to the downsampling factors.

TABLE 1 Acquisition parameters in our study for the synthetic and *in vivo* 4D-flow MRI data set.

	Synthetic cerebrovascular	*In vivo* Cerebrovascular	*In vivo* Cardiovascular
FOV [mm]	320–440 x 370–520 x 320–470	190 x 210 x 32	340–360 x 210–250 x 80–150
Acquisition matrix	-	128 x 128 x 32	160 x 100 x 32–64
Spatial res. [mm]	0.25 x 0.25 x 0.25	0.82 x 0.82 x 0.82	2.1–2.3 x 2.1–2.5 x 2.3–2.5
Temporal res. [ms]	36	55	40
Patient cohort	6	3	21
Scanner	-	3.0 T Philips Achieva TX	1.5 T Siemens avanto
TE/TR [ms]	-	2.9/4.6	2.54/5
V_{enc} [cm/s]	-	80	150
Parallel imaging	-	SENSE ($R = 2$) (Pruessmann et al., 1999)	PEAK-GRAPPA ($R = 5$) (Jung et al., 2008)
Cardiac gating	-	peripheral pulse unit	ECG

FOV, field of view; TE, echo time; TR, repetition time; V_{enc}, velocity encoding range; ECG, electrocardiogram.

3. We apply additive Gaussian noise with the k-space.
4. Finally, we apply Fourier inversion and multiply the phase with *venc* to obtain the low-resolution training data.

This process closely resembles the sub-sampling process in the MRI scanner (Gudbjartsson and Patz, 1995). For our training, we extract patches of sizes $24\times24\times12\times3$, $16\times16\times8\times3$, and $12\times12\times6\times3$ from the low-resolution volume for $2\times$, $3\times$, and $4\times$ super-resolution, respectively. The patches-size corresponding to the high-resolution volume is $48\times48\times24\times3$. The patches are selected on-the-fly during training from a random location of the training data. We normalize the input to [-1,1] by dividing the velocity with v_{max}, the maximum velocity present.

2.6. Evaluation metrics

Peak velocity-to-noise ratio (PVNR) is commonly used to quantify the reconstruction quality of the estimated velocity field. We use PVNR as a primary metric to quantify the performance of our proposed super-resolution method. PVNR between reference ($\mathbf{u}$) and the estimated velocity ($\mathbf{v}$) is

$$\text{PVNR} = 20\log_{10}\frac{1}{\text{RMS}_{vel}}\,\text{dB} \tag{6}$$

$$\text{RMS}_{vel} = \frac{1}{\max_i \|\mathbf{u}_i\|}\sqrt{\frac{1}{N}\sum_{i=0}^{N}\|\mathbf{u}_i - \mathbf{v}_i\|^2} \tag{7}$$

where RMS_{vel} is the normalized-root-mean-squared-error of velocity. PVNR represents a combined error in the magnitude estimation and the phase estimation. Furthermore, we aim to deconstruct the source of error into its magnitude and phase component. As a measure of the error in magnitude estimation, we compare the normalized-root-mean-squared-error of speed (RMS_{speed}) as described below

$$\text{RMS}_{speed} = \frac{1}{\max_i \|\mathbf{u}_i\|}\sqrt{\frac{1}{N}\sum_{i=0}^{N}(\|\mathbf{u}_i\| - \|\mathbf{v}_i\|)^2} \tag{8}$$

We also compute Direction Error ($\mathcal{E}_{dir}$) to measure the deviation of instantaneous velocity direction with respect to the reference velocity. The error in direction estimation is critical in some downstream tasks, such as streamline-tracing and path-line tracking, where the corresponding algorithm's accuracy depends on direction estimation accuracy.

$$\mathcal{E}_{dir} = \frac{1}{N}\sum_{i=0}^{N}\left(1 - \frac{\langle\mathbf{u}_i\cdot\mathbf{v}_i\rangle}{\|\mathbf{u}_i\|\,\|\mathbf{v}_i\|}\right) \tag{9}$$

Furthermore, we emphasize the flow consistency in terms of the flow divergence of the super-resolved flow field. We compute the root-mean-squared divergence in the region of interest and compare it against the high-resolution reference velocity.

$$\text{RMS}_{div} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}|\nabla\cdot\mathbf{v}_i|^2} \tag{10}$$

3. Results

In this section, we describe our experiments and the main results. We refer to WDSR-3D as the 3D extension of WDSR by Yu et al. (2018). For the details of WDSR architecture, please refer to the original paper by Yu et al. (2018). This is one of the top-performing methods in the NTIRE super-resolution challenge. Hence, we select this as the baseline of our study and build our contribution upon it. We compare our work to the WDSR-3D for two main purposes. First, it is a strong baseline that is scalable to 3D. Second, its residual learning architecture is similar to the existing method (Ferdian et al., 2020) and

provides a point of reference for comparison. The method by Ferdian et al. (2020) also requires magnitude images, which is not a good candidate for training with CFD simulated data. Since we are not using magnitude images in our training, we are unable to perform a direct comparison. Although it is not 100% identical, WDSR-3D is analogous to their method and can serve as a point of reference. In our experiment with WDSR-3D on synthetic data, we observe that ℓ_1 loss offers on average 1dB PVNR improvement over ℓ_2 loss for 2x super-resolution. From this observation, we chose WDSR-3D with ℓ_1 loss as the baseline model for our experiments. We will denote SRflow (ℓ_1), SRflow (mp-ℓ_1), and SRflow (opt) trained with $\mathcal{J}_{\ell_1}$, $\mathcal{J}_{mp\text{-}\ell_1}$, and $\mathcal{J}_{opt}$, respectively. The following two subsections (Experiment-1 & -2) are the descriptions of the experimental setup. For statistical significance analysis, we performed a Wilcoxon signed-rank test. For this, we collected predictions from all of the 3-fold validation. We declare statistical significance when the p-value is lower than 0.001. The analysis of their results is presented jointly in Section 4.

3.1. Experiment-1: Train on synthetic cerebrovascular data

First, we train and test our model on the synthetic cerebrovascular data. We perform three experiments with three different train-validation splits and report the combined results. This allows us to maximize training examples while performing required cross-validation. For each experiment, we have selected (120 samples) five subjects as training and the remaining one as the validation data (24 samples). The train-validation split was fixed across the models and loss functions for a fair comparison.

3.1.1. Part A: Evaluation on synthetic cerebrovascular data

This experiment serves as the proof of concept for both our model and loss function. We compute each metric's mean and SD for the validation data over three independent trials. We train separate models for three different upscaling factors, such as 2×, 3×, and 4× SR. Figure 5A shows the boxplot of four different metrics for the experiments on the synthetic data. We present the comparative result for different experiments set in Supplementary Table S1. Supplementary Figure S1 presents an exemplary temporal visualization of PVNR for a particular slice from the validation set. Supplementary Figure S2 presents the error profile of the velocity field for the corresponding slice location of the same example, which is done using Paraview.

3.1.2. Part B: Evaluation on *in vivo* cerebrovascular data

We evaluate the model trained on synthetic cerebrovascular data on the 32 *in vivo* cerebrovascular samples from 2 subjects. Supplementary Table S2 shows the quantitative comparison of all the metrics for three different scaling factors. We observe that the improvement in metrics for scaling factors 2× and 3× is low compared to 4×. The improvement is also relatively lower than the improvement observed in Supplementary Table S1. Although SRflow (opt) consistently performs better than the cubic spline and baseline WDSR-3D, we investigate the inclusion of *in vivo* data during training in the following.

3.2. Experiment-2: Fine-tune on *in vivo* cardiovascular data

The previous experiment shows that the model trained on synthetic data does not offer the same degree of improvement over cubic-spline on *in vivo* cerebrovascular data for lower scaling factors. We attribute this to the fact that different noises and artifacts are present in the *in vivo* data (Johnson and Markl, 2010). We fine-tune the model using *in vivo* cardiovascular data to overcome this gap. We choose to fine-tune all our model Experiment-1 on *in vivo* cardiovascular data instead of *in vivo* cerebrovascular data because cardiovascular data have more samples than cerebrovascular data and consists of a significantly richer variation of noise and artifacts (Fathi et al., 2018). Similar to synthetic experiments, we perform three experiments with three different train-validation splits and report the combined results. For each experiment, we have selected 17 subjects as training and the remaining 4 as the validation set. Similar to before, the train-validation split was fixed across the models and loss functions for a fair comparison. Finally, we directly translate the trained model from cardiovascular data to *in vivo* aneurysm 4D-flow MRI data and evaluate the performance.

3.2.1. Part A: Evaluation of *in vivo* cardiovascular data

Similar to our synthetic data experiments, we perform the experiments for three different scenarios, such as 2×, 3×, and 4× super-resolution. Figure 5B shows the boxplot of four different metrics for the validation data. The comparative result for this experiment is shown in Supplementary Table S3. Figure 6 shows a qualitative comparison of representative cardiac data from our experiments.

3.2.2. Part B: Re-evaluation of *in vivo* cerebrovascular data

We use the trained model from the cardiovascular experiments and evaluate the *in vivo* cerebrovascular data

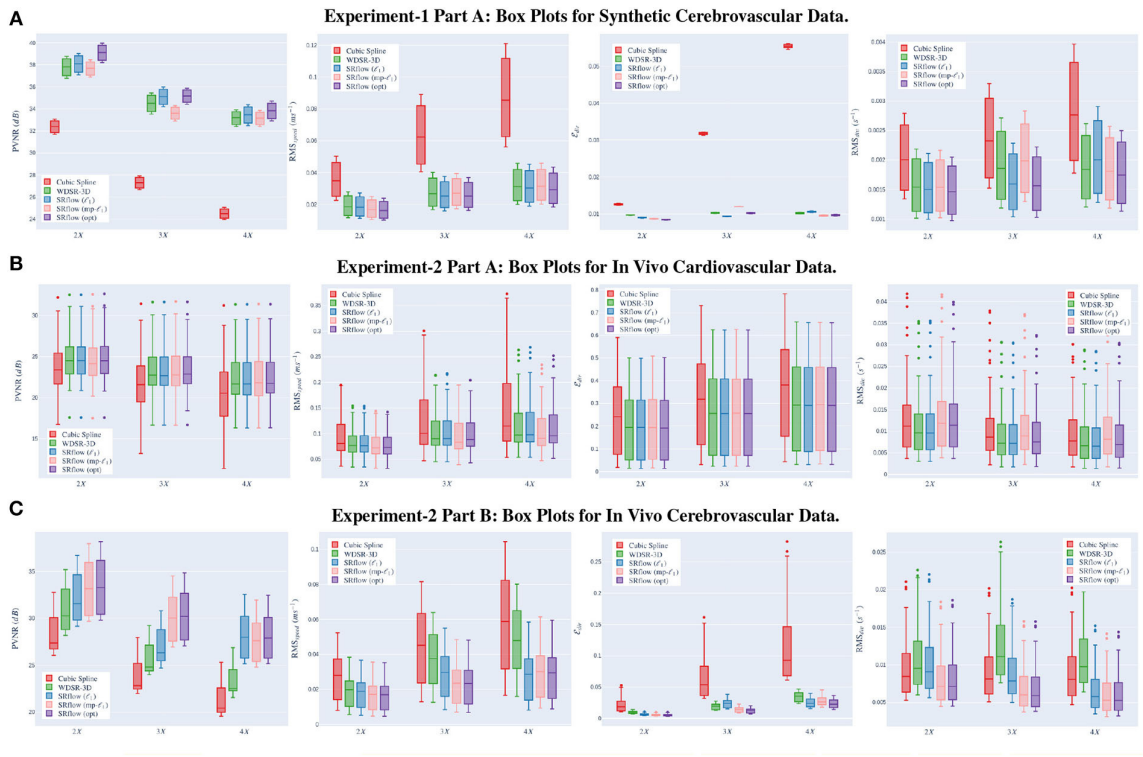

FIGURE 5
Left → right shows the box plots for peak velocity-to-noise ratio (PVNR), root-mean-squared-error $(RMS)_{speed}$, $\mathcal{E}_{dir}$, and RMS_{div}, respectively. The x-axis shows different upscaling factors 2×, 3×, and 4× respectively. Higher (↑) PVNR and lower (↓) RMS_{speed}, $\mathcal{E}_{dir}$ and RMS_{div} indicates better performance. Note that SRflow (opt) consistently outperforms the baseline models. **(A)** Experiment-1 Part A: box plots for synthetic cerebrovascular data. **(B)** Experiment-2 Part A: box plots for *in vivo* cardiovascular data. **(C)** Experiment-2 Part B: box plots for *in vivo* cerebrovascular data.

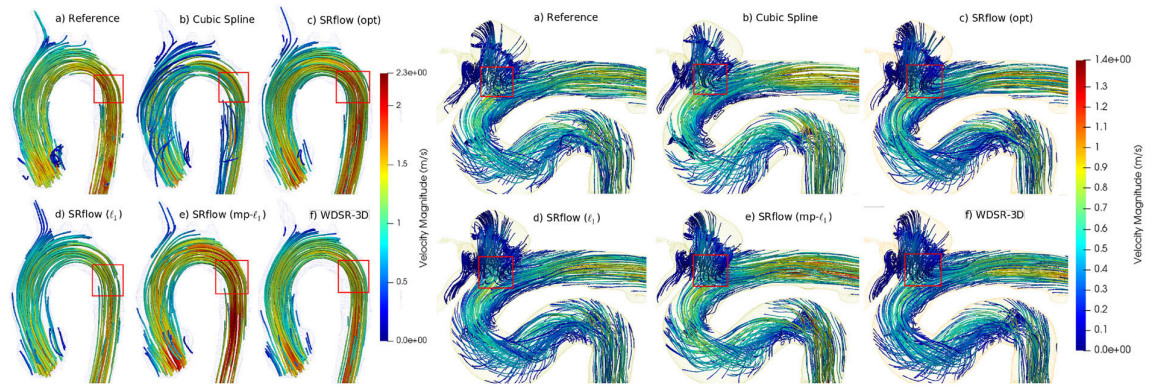

FIGURE 6
Qualitative comparison of streamlines obtained from different methods on two representative *in vivo* samples (left-cardiovascular and right-cerebrovascular). SRflow (opt) shows closer similarity to the reference streamlines for both the cases, while it reduces artifacts.

without further fine-tuning. Supplementary Table S4 shows the quantitative comparison of all the metrics for three different scaling factors. Figure 5C shows the boxplot of four different metrics for the *in vivo* cerebrovascular data. Figure 6 shows a qualitative comparison of representative aneurysm data from our experiments. Additionally, we have shown the

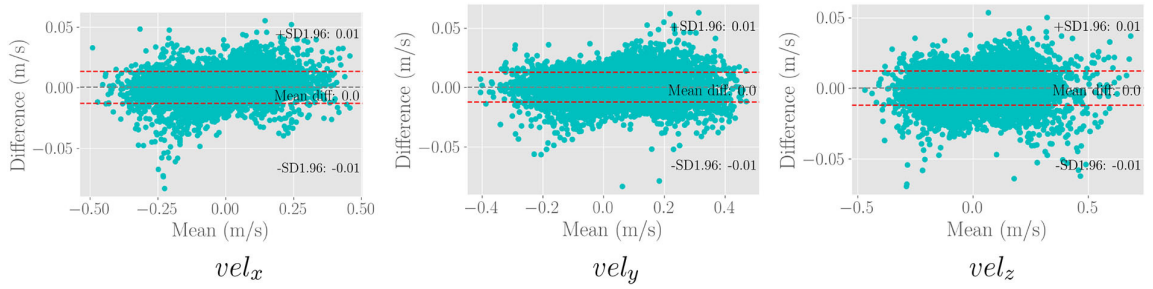

FIGURE 7
Bland-Altman plot for the three velocity components of 50,000 random samples from the *in vivo* cerebrovascular data for the SRflow(opt) model fine-tuned with cardiovascular data. We observe excellent agreement between the prediction and the reference.

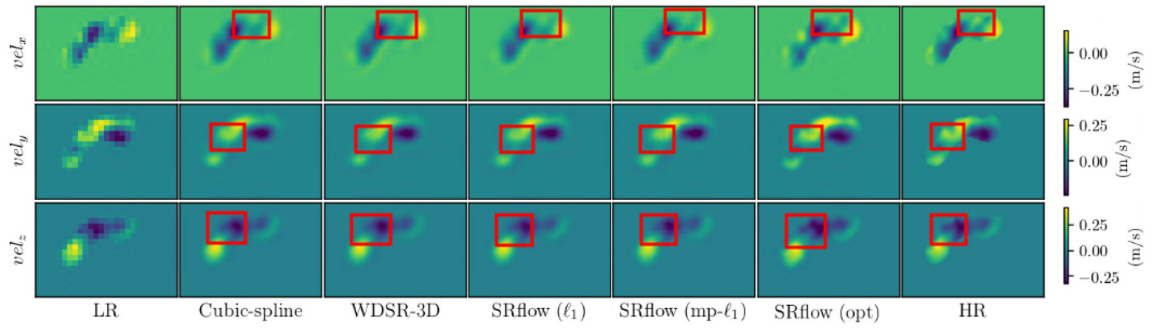

FIGURE 8
Qualitative images from different methods on a representative cerebrovascular sample. Note that cubic spline interpolation creates unnecessary amplification and attenuation of flow where SRflow (opt) preserves the flow intensity closest to the reference (HR).

Bland-Altman plot (Figure 7) for each velocity component, which shows good agreement with the reference, and the error is homogeneously scattered across the (mean $\pm 1.96 \times$ SD) range. Qualitative results from Figure 8 show that SRflow (opt) produces the closest prediction to the reference.

4. Discussion

4.1. Ablation study and transfer learning

4.1.1. Effect of squeeze and excite block

From Figures 5A–C and Supplementary Tables S1–S4 in the Appendix, we observe that cubic-spline-based upsampling is consistently inferior compared to the learning-based solutions. The performance gain between these two increases with the upsampling factor for both the synthetic and *in vivo* data. However, the improvement for synthetic cerebrovascular data (Supplementary Table S1) is greater than the *in vivo* cardiovascular data (Supplementary Table S3). We attribute this to the 'reference' 4D-flow MRI data being noisy, which may hamper the reconstruction quality measure. Furthermore, we notice that the SRflow (ℓ_1) network achieves better PVNR, RMS_{speed}, $\mathcal{E}_{dir}$, and RMS_{div}. than the WDSR-3D counterpart for all three cases of super resolutions $2\times$, $3\times$, and $4\times$. We find SRflow (ℓ_1) results are statistically significant ($p < 0.001$) compared to WDSR-3D for all four metrics. We believe this is due to the fact that the diversity in feature space induced by the SE block better captures the inter-scale relationship of the velocity field.

4.1.2. Effect of our proposed loss function (*mp-*ℓ_1)

We investigate the effect of two different loss functions discussed in Section 2.2 on the SRflow architecture. For Experiment 1 (Supplementary Tables S1, S2), we see that SRflow (opt) performs consistently better than the other loss functions for all super-resolution factors concerning PVNR and RMS_{speed}. For $\mathcal{E}_{dir}$, SRflow (opt) produces the lowest error except $4\times$ factor in Supplementary Table S1. For RMS_{div}, we observe that SRflow (ℓ_1) produces the lowest error. We find the improvements

TABLE 2 Effect of divergence loss and data augmentation strategy on synthetic cerebrovascular data.

Methods	s	PVNR (dB) ↑	RMS_{speed} (ms^{-1}) ↓	$\mathcal{E}_{dir}$ ↓	RMS_{div} (s^{-1}) ↓
SRflow (opt)	×2	39.14 ± 0.629	0.0161 ± 0.00487	0.0084 ± 0.00008	0.0014 ± 0.00038
SRflow ($\ell_1 + div$)	×2	38.89 ± 0.427	**0.0138 ± 0.00346**	**0.0081 ± 0.00005**	**0.0012 ± 0.00026**
SRflow (opt+mixed data)	×2	39.21 ± 0.301	**0.0138 ± 0.00249**	0.0084 ± 0.00008	0.0013 ± 0.00028
SRflow (opt + div)	×2	**38.93 ± 0.426**	**0.0138 ± 0.00342**	0.0084 ± 0.00007	**0.0012 ± 0.00027**
SRflow (opt)	×3	35.20 ± 0.520	0.0253 ± 0.00732	0.0102 ± 0.00008	0.0015 ± 0.00042
SRflow ($\ell_1 + div$)	×3	35.45 ± 0.376	0.0205 ± 0.00494	**0.0088 ± 0.00006**	0.0013 ± 0.00029
SRflow (opt+mixed data)	×3	35.19 ± 0.260	0.0241 ± 0.00405	0.0102 ± 0.00005	0.0015 ± 0.00026
SRflow (opt + div)	×3	**35.94 ± 0.389**	**0.0196 ± 0.00478**	**0.0088 ± 0.00006**	**0.0012 ± 0.00028**
SRflow (opt)	×4	33.87 ± 0.642	0.0293 ± 0.00888	0.0097 ± 0.00015	0.0017 ± 0.00048
SRflow ($\ell_1 + div$)	×4	34.44 ± 0.399	**0.0226 ± 0.00560**	**0.0096 ± 0.00006**	0.0014 ± 0.00032
SRflow (opt+mixed data)	×4	33.24 ± 0.256	0.0269 ± 0.00603	**0.0096 ± 0.00002**	**0.0013 ± 0.00030**
SRflow (opt + div)	×4	**34.51 ± 0.428**	0.0229 ± 0.00573	0.0098 ± 0.00003	**0.0013 ± 0.00031**

The bold values mean the highest score of the corresponding scale factor ×2, ×3, and ×4.

TABLE 3 Comparison on data augmentation strategy on *in vivo* cerebrovascular data.

Methods	s	PVNR (dB) ↑	RMS_{speed} (ms^{-1}) ↓	$\mathcal{E}_{dir}$ ↓	RMS_{div} (s^{-1}) ↓
SRflow (opt)	×2	**33.52 ± 2.703**	**0.0164 ± 0.00878**	0.0053 ± 0.00160	**0.0083 ± 0.00394**
SRflow (opt+mixed data)	×2	33.32 ± 2.588	0.0167 ± 0.00878	**0.0051 ± 0.00131**	0.0083 ± 0.00387
SRflow (opt)	×3	**30.46 ± 2.473**	**0.0228 ± 0.01188**	**0.0120 ± 0.00345**	**0.0070 ± 0.00333**
SRflow (opt+mixed data)	×3	30.39 ± 2.390	0.0229 ± 0.01177	0.0122 ± 0.00347	0.0070 ± 0.00336
SRflow (opt)	×4	**28.30 ± 2.321**	**0.0279 ± 0.01456**	**0.0242 ± 0.00723**	0.0067 ± 0.00325
SRflow (opt+mixed data)	×4	28.03 ± 2.258	0.0294 ± 0.01491	0.0245 ± 0.00694	**0.0062 ± 0.00311**

The bold values mean the highest score of the corresponding scale factor ×2, ×3, and ×4.

from SRflow (opt) over baseline (WDSR-3D) and other SRflow variants to be statistically significant ($p < 0.001$) for PVNR, RMS_{speed}, and $\mathcal{E}_{dir}$. While we do not find any statistically significant ($p > 0.001$) difference between the SRflow (ℓ_1) and SRflow (opt) for RMS_{div} for S1, the same is statistically significant ($p < 0.001$) for Supplementary Table S2.

For experiment 2 (Supplementary Tables S3, S4), SRflow (opt) improves PVNR for all scale factors compared to baseline (WDSR-3D) and other SRflow variants. For RMS_{speed}, SRflow (opt) and SRflow (mp-ℓ_1) result in the lowest error. For $\mathcal{E}_{dir}$, SRflow (opt) produces the lowest error except for the 4× factor in Supplementary Table S1. WDSR-3D and SRflow (ℓ_1) produce the lowest RMS_{div}. Although SRflow (opt) reduces $\mathcal{E}_{dir}$ consistently, it produces slightly higher RMS_{speed} and RMS_{div} than SRflow (mp-ℓ_1) and SRflow (ℓ_1), respectively. We attribute this to the fact that mp-ℓ_1 offers a trade-off between accurate magnitude and phase estimation of the velocity field during training on 'noisy reference.' While higher PVNR ensures good signal quality for accurate quantitative analysis, lower $\mathcal{E}_{dir}$ reduces error in the qualitative assessment, such as streamline tracing. We find the improvements from SRflow (opt) over baseline (WDSR-3D) and other SRflow variants to be statistically significant ($p < 0.001$) for PVNR, RMS_{speed}, and $\mathcal{E}_{dir}$. We find no statistically significant ($p < 0.001$) difference between the SRflow (ℓ_1) and SRflow (opt) for RMS_{div} for Supplementary Tables S3, S4. In stark contrast with Supplementary Table S3, this shows the benefit of fine-tuning using *in vivo* data.

The mp-ℓ_1 alone fails to improve the performance compared to the standard ℓ_1, but it is evident from Supplementary Tables S1, S3 that when combined with the ℓ_1 it outperforms both. We hypothesize that gradient from mp-ℓ_1 loss is beneficial when the directional error is large because of the directional sensitivity, which is helpful for the 'exploration' in the early stage of training. Additionally, since the loss curve of mp-ℓ_1 is smoother compared to ℓ_1 loss at lower error, ℓ_1 provides a stronger gradient than mp-ℓ_1, which is vital in the 'exploitation' of the final stage of training. Hence, the performance improvement stands out when both loss functions are used simultaneously.

We experimented with divergence loss function, which can serve as a physics constrained regularizer. The divergence loss is

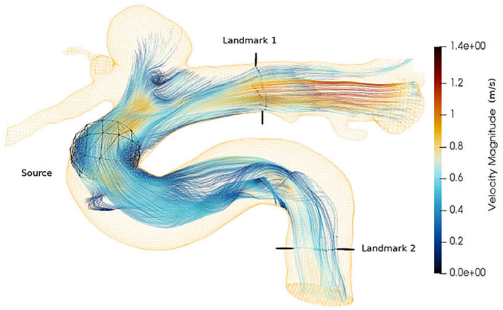

FIGURE 9
Velocity streamlines from an *in vivo* cerebrovascular 4D-flow MRI. The seed points are placed using a spherical source. The landmarks are used in Table 4.

defined below.

$$\mathcal{J}_{\text{div}} = \frac{1}{n}\sum_{i=1}^{n} \|\nabla \cdot \mathbf{v}_i\|_2 \tag{11}$$

We have identical experiment settings for divergence loss as described before to have a fair comparison. We have observed from Table 2 that the inclusion of divergence loss improved the results in both cases. In particular, the divergence metric improved consistently across the different upsampling factors. We also observe that the gain is slightly higher in the case of SRflow (opt) than in standard ℓ_1 loss, which reasserts the effectiveness of our proposed loss function. However, since divergence cannot be used as a standalone loss function, we conclude from these experiments that using it as a regularizer benefits the model.

4.1.3. Transfer learning

We fine-tune our model on the *in vivo* cardiovascular data and validate it on the *in vivo* cerebrovascular data, scanned in an entirely different scanner and velocity encoding value. Despite the difference in the anatomical region, we observe that our proposed SRflow (opt) improves all metrics significantly (c.f. Supplementary Table S4). Especially we observe that it produces the lowest RMS_{div} compared to other methods, which is also statistically significant. This observation confirms that our model can be seamlessly transferred to other existing MRI acquisition configurations without the need for any further local fine-tuning.

We have performed additional experiments comprising joint CFD and *in vivo* datasets and reported Tables 2, 3. We compare the result from joint training and training sequentially in synthetic and *in vivo* data on both synthetic and *in vivo* data. We observe a marginal improvement on the synthetic test set. However, the results deteriorate slightly on the *in vivo* test set.

TABLE 4 The mean of relative error in the number of streamlines with respect to the reference *in vivo* data over one cardiac cycle, which is computed at two landmarks as shown in Figure 9 for different super-resolution methods.

Method	Landmark 1	Landmark 2
Cubic-spline	0.66	0.76
WDSR-3D	0.50	0.16
SRflow (ℓ_1)	0.37	0.16
SRflow (mp-ℓ_1)	0.36	0.21
SRflow (opt)	0.31	0.11

TABLE 5 Runtime comparison between SRflow and cubic spline, where we see that SRflow is much faster.

Method	2x	3x	4x
Cubic spline	76.42 s	76.19 s	73.09 s
SRflow (opt)	8.17 s	3.80 s	2.86 s

We attribute this to the fact that the MRI artifacts and noise are difficult to model in the CFD data.

4.2. Global quality evaluation of reconstructed velocity

Besides the voxel-wise reconstruction metrics, analyzing the effect of the super-resolved velocity field on a global level, such as path integration along the flow field, are also important. We compare the computed streamlines for *in vivo* 4D-flow MRI data to assess the global reconstruction quality.

Streamlines (Cebral et al., 2011) is an important visualization technique, often used as the primary mode for clinicians' interpretable representation of 4D-flow MRI. The continuity of streamlines can be used as an alternative way to measure the quality of the super-resolved vector fields. We start the streamline tracing near the aneurysm with 2000 seed points and extend toward both the landmarks as shown in Figure 9. We compute the relative error between the number of streamlines produced by each super-resolver volume and the reference streamline. Table 4 shows the mean of the relative error over one cardiac cycle at two landmarks for the 4× super-resolution task. We find that the cubic spline always underestimates the number of streamlines, and SRflow (opt) produces the lowest relative errors.

4.3. Runtime comparison

We compare the runtime for a 96 × 132 × 48 3D volume between the cubic spline and the neural network-based model for three different resolutions. We compare the runtime in a

workstation equipped with Intel(R) Xeon(R) W-2123 and 64 GB DDR4 RAM. The comparison is shown in Table 5. SRflow offers significant computational speedup, which is favorable for clinical application.

4.4. Limitation and outlook

While the improved resolution will be beneficial in increased stability for numerical gradient computation, its accuracy is still limited to finite difference schemes and the maximum super-resolution factor learned during training for optimal performance. Furthermore, the current study is limited in exploring different spatial super-resolution factors, and temporal super-resolution is of future research interest. Additionally, including other realistic perturbations, such as scan-rescan variability, phase aliasing, and eddy current effect, would be of interest to include in the model. Future research will include increasing the number of samples of the *in vivo* cohort. Future work will also focus on further quantitative assessment of advanced parameters, such as WSS and KE.

5. Conclusion

This paper investigates the effectiveness of deep learning in super-resolving 4D-flow MRI data up to 4x resolution. We have started with a strong baseline model and gradually improved it by incorporating expressive squeeze and excite block. Furthermore, we introduce a novel robust loss function with directional sensitivity suitable for velocity data. With extensive validation, we demonstrated the effectiveness of the introduced component. Next, we show that the model learned from synthetic CFD data still requires finetuning on *in vivo* data for improved performance. Importantly, this finetuning is not dataset-dependant and can be applied seamlessly to other *in vivo* datasets without further finetuning. Naturally, we have improved runtime compared to the classical interpolation method, which could enable future 4D-flow MRI acquisitions at lower resolution—and thus with decreased scan time—without compromising the accuracy of quantitative flow analysis.

Data availability statement

The datasets presented in this article are not readily available because restrictions apply to the sharing of patient data that supports the findings of this study. Requests to access the datasets should be directed to SS, suprosanna.shit@tum.de.

Ethics statement

The studies involving human participants were reviewed and approved by Institutional Ethics Committee, TUM. The patients/participants provided their written informed consent to participate in this study.

Author contributions

SS and JZ conceived and designed the study and performed the experiments. JZ and AS acquired the datasets. AS simulated the synthetic blood flow. SS, JZ, and AS preprocessed the dataset. SS, JZ, IE, JP, and CP analyzed the results. SS, JZ, CP, and BM prepared the manuscript. BM supervised the study. All authors contributed to the article and approved the submitted version.

Funding

SS and IE are supported by the Translational Brain Imaging Training Network under the EU Marie Sklodowska-Curie programme (Grant ID: 765148).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2022.928181/full#supplementary-material

References

Albay, E., Demir, U., and Unal, G. (2018). "Diffusion MRI spatial super-resolution using generative adversarial networks," in *Proceedings of PRIME Workshop MICCAI* (Springer), 155–163.

Amili, O., Schiavazzi, D., Moen, S., Jagadeesan, B., Van de Moortele, P.-F., et al. (2018). Hemodynamics in a giant intracranial aneurysm characterized by *in vitro* 4D flow MRI. *PLoS ONE* 13, e0188323. doi: 10.1371/journal.pone.0188323

Bakhshinejad, A., Baghaie, A., Vali, A., Saloner, D., Rayz, V. L., and DSouza, R. M. (2017). Merging computational fluid dynamics and 4D Flow MRI using proper orthogonal decomposition and ridge regression. *J. Biomech.* 58, 162–173. doi: 10.1016/j.jbiomech.2017.05.004

Bernstein, M. A., Fain, S. B., and Riederer, S. J. (2001). Effect of windowing and zero-filled reconstruction of MRI data on spatial resolution and acquisition strategy. *JMRI* 14, 270–280. doi: 10.1002/jmri.1183

Bhowmik, A., Shit, S., and Seelamantula, C. S. (2018). Training-free, single-image super-resolution using a dynamic convolutional network. *IEEE Signal Process. Lett.* 25, 85–89. doi: 10.1109/LSP.2017.2752806

Breuer, F. A., Kellman, P., Griswold, M. A., and Jakob, P. M. (2005). Dynamic autocalibrated parallel imaging using temporal grappa (TGRAPPA). *Magn. Reson. Med.* 53, 981–985. doi: 10.1002/mrm.20430

Brindise, M. C., Rothenberger, S., Dickerhoff, B., Schnell, S., Markl, M., Saloner, D., et al. (2019). Multi-modality cerebral aneurysm haemodynamic analysis: in vivo 4D flow MRI, *in vitro* volumetric particle velocimetry and in silico computational fluid dynamics. *J. R. Soc. Interface* 16, 20190465. doi: 10.1098/rsif.2019.0465

Byrne, G., Mut, F., and Cebral, J. (2014). Quantifying the large-scale hemodynamics of intracranial aneurysms. *AJNR Am. J. Neuroradiol.* 35, 333–338. doi: 10.3174/ajnr.A3678

Callaghan, F. M., and Grieve, S. M. (2017). Spatial resolution and velocity field improvement of 4D-flow MRI. *Magn. Reson. Med.*, 78, 1959–1968. doi: 10.1002/mrm.26557

Cebral, J. R., Mut, F., Weir, J., and Putman, C. M. (2011). Association of hemodynamic characteristics and cerebral aneurysm rupture. *AJNR Am. J. Neuroradiol.* 32, 264–270. doi: 10.3174/ajnr.A2274

Chen, Y., Shi, F., Christodoulou, A. G., Xie, Y., Zhou, Z., and Li, D. (2018). "Efficient and accurate MRI super-resolution using a generative adversarial network and 3D multi-level densely connected network," in *Proceedings of MICCAI* (Springer), 91–99.

Chu, M., Xie, Y., Leal-Taixé, L., and Thuerey, N. (2018). Temporally coherent gans for video super-resolution (tecogan). *CoRR, abs/1811.09393*.

Cibis, M., Bustamante, M., Eriksson, J., Carlhäll, C.-J., and Ebbers, T. (2017). Creating hemodynamic atlases of cardiac 4D flow MRI. *J. Magn. Reson. Imaging* 46, 1389–1399. doi: 10.1002/jmri.25691

Dong, C., Loy, C. C., He, K., and Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern. Anal. Mach. Intell.* 38, 295–307. doi: 10.1109/TPAMI.2015.2439281

Dyverfeldt, P., Ebbers, T., and Länne, T. (2014). Pulse wave velocity with 4D flow MRI: systematic differences and age-related regional vascular stiffness. *Magn. Reson. Med.* 32, 1266–1271. doi: 10.1016/j.mri.2014.08.021

Fathi, M. F., Bakhshinejad, A., Baghaie, A., Saloner, D., Sacho, R. H., Rayz, V. L., et al. (2018). Denoising and spatial resolution enhancement of 4D flow MRI using proper orthogonal decomposition and lasso regularization. *Comput. Med. Imaging Graph.* 70, 165–172. doi: 10.1016/j.compmedimag.2018.07.003

Fathi, M. F., Perez-Raya, I., Baghaie, A., Berg, P., Janiga, G., Arzani, A., et al. (2020). Super-resolution and denoising of 4D-Flow MRI using physics-informed deep neural nets. *Comput. Methods Programs Biomed.* 197, 105729. doi: 10.1016/j.cmpb.2020.105729

Ferdian, E., Suinesiaputra, A., Dubowitz, D. J., Zhao, D., Wang, A., Cowan, B., et al. (2020). 4DFlowNet: Super-resolution 4D Flow MRI using deep learning and computational fluid dynamics. *Front. Phys.* 8, 138. doi: 10.3389/fphy.2020.00138

Futami, K., Uno, T., Misaki, K., Tamai, S., Nambu, I., Uchiyama, N., et al. (2019). Identification of vortex cores in cerebral aneurysms on 4D flow MRI. *AJNR Am. J. Neuroradiol.* 40, 2111–2116. doi: 10.3174/ajnr.A6322

Gaidzik, F., Pathiraja, S., Saalfeld, S., Stucht, D., Speck, O., Thévenin, D., et al. (2021). Hemodynamic data assimilation in a subject-specific circle of willis geometry. *Clin. Neuroradiol.* 31, 643–651. doi: 10.1007/s00062-020-00959-2

Gaidzik, F., Stucht, D., Roloff, C., Speck, O., Thévenin, D., and Janiga, G. (2019). Transient flow prediction in an idealized aneurysm geometry using data assimilation. *Comput. Biol. Med.* 115, 103507. doi: 10.1016/j.compbiomed.2019.103507

Garcia, J., Barker, A. J., and Markl, M. (2019). The role of imaging of flow patterns by 4D flow MRI in aortic stenosis. *JACC Cardiovasc. Imaging* 12, 252–266. doi: 10.1016/j.jcmg.2018.10.034

Gudbjartsson, H., and Patz, S. (1995). The rician distribution of noisy MRI data. *Magn. Reson. Med.* 34, 910–914. doi: 10.1002/mrm.1910340618

Guzzardi, D. G., Barker, A. J., Van Ooij, P., Malaisrie, S. C., Puthumana, J. J., Belke, D. D., et al. (2015). Valve-related hemodynamics mediate human bicuspid aortopathy: insights from wall shear stress mapping. *JACC Cardiovasc. Imaging* 66, 892–900. doi: 10.1016/j.jacc.2015.06.1310

Habibi, M., D'Souza, R. M., Dawson, S. T., and Arzani, A. (2021). Integrating multi-fidelity blood flow data with reduced-order data assimilation. *Comput. Biol. Med.* 135, 104566. doi: 10.1016/j.compbiomed.2021.104566

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of CVPR*, 7132–7141.

Johnson, K. M., and Markl, M. (2010). Improved SNR in phase contrast velocimetry with five-point balanced flow encoding. *Magn. Reson. Med.* 63, 349–355. doi: 10.1002/mrm.22202

Jung, B., Ullmann, P., Honal, M., Bauer, S., Hennig, J., and Markl, M. (2008). Parallel MRI with extended and averaged GRAPPA kernels (PEAK-GRAPPA): optimized spatiotemporal dynamic imaging. *J. Magn. Reson. Imaging* 28, 1226–1232. doi: 10.1002/jmri.21561

Kim, J., Lee, J. K., and Lee, K. M. (2016). "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of CVPR*, 1637–1645.

Klemens, F., Schuhmann, S., Balbierer, R., Guthausen, G., Nirschl, H., Thäter, G., et al. (2020). Noise reduction of flow MRI measurements using a lattice boltzmann based topology optimisation approach. *Comput. Fluids* 197, 104391. doi: 10.1016/j.compfluid.2019.104391

Leidenberger, T., Gordon, Y., Farag, M., Delles, M., Sanches, A. F., Fink, M. A., et al. (2020). Imaging-based 4D aortic pressure mapping in Marfan syndrome patients: a matched case-control study. *Ann. Thorac Surg.* 109, 1434–1440. doi: 10.1016/j.athoracsur.2019.08.048

Li, B., and Han, L. (2013). "Distance weighted cosine similarity measure for text classification," in *Proceedings of IDEAL* (Springer), 611–618.

Lim, B., Son, S., Kim, H., Nah, S., and Mu Lee, K. (2017). "Enhanced deep residual networks for single image super-resolution," in *Proceedings of CVPR Workshops*, 136–144.

Liu, P., Lyu, M., King, I., and Xu, J. (2019). "Selflow: self-supervised learning of optical flow," IN *Proceedings of CVPR*, 4571–4580.

Lyu, Q., Shan, H., Steber, C., Helis, C., Whitlow, C., Chan, M., et al. (2020). Multi-contrast super-resolution mri through a progressive network. *IEEE Trans. Med. Imaging* 39, 2738–2749. doi: 10.1109/TMI.2020.2974858

Ma, L. E., Markl, M., Chow, K., Huh, H., Forman, C., Vali, A., et al. (2019). Aortic 4D flow MRI in 2 minutes using compressed sensing, respiratory controlled adaptive k-space reordering, and inline reconstruction. *Magn. Reson. Med.* 81, 3675–3690. doi: 10.1002/mrm.27684

Markl, M., Frydrychowicz, A., Kozerke, S., Hope, M., and Wieben, O. (2012). 4D flow MRI. *J. Magn. Reson. Imaging* 36, 1015–1036. doi: 10.1002/jmri.23632

Meurée, C., Maurel, P., Ferré, J.-C., and Barillot, C. (2019). Patch-based super-resolution of arterial spin labeling magnetic resonance images. *Neuroimage* 189, 85–94. doi: 10.1016/j.neuroimage.2019.01.004

Mura, J., Pino, A. M., Sotelo, J., Valverde, I., Tejos, C., Andia, M. E., et al. (2016). Enhancing the velocity data from 4D flow MR images by reducing its divergence. *IEEE Trans. Med. Imaging* 35, 2353–2364. doi: 10.1109/TMI.2016.2570010

Nguyen, H. V., and Bai, L. (2010). "Cosine similarity metric learning for face verification," in *Proceedings of ACCV* (Springer), 709–720.

Ong, F., Uecker, M., Tariq, U., Hsiao, A., Alley, M. T., Vasanawala, S. S., et al. (2015). Robust 4D flow denoising using divergence-free wavelet transform. *Magn. Reson. Med.* 73, 828–842. doi: 10.1002/mrm.25176

Petersson, S., Dyverfeldt, P., and Ebbers, T. (2012). Assessment of the accuracy of MRI wall shear stress estimation using numerical simulations. *J. Magn. Reson. Imaging* 36, 128–138. doi: 10.1002/jmri.23610

Pham, C.-H., Ducournau, A., Fablet, R., and Rousseau, F. (2017). "Brain MRI super-resolution using deep 3D convolutional networks," in *Proceedings of ISBI* (Melbourne, VIC: IEEE), 197–200.

Pruessmann, K. P., Weiger, M., Scheidegger, M. B., and Boesiger, P. (1999). Sense: sensitivity encoding for fast MRI. *Magn. Reson. Med*. 42, 952–962. doi: 10.1002/(SICI)1522-2594(199911)42:5andlt;952::AID-MRM16andgt;3.0.CO;2-S

Rutkowski, D. R., Roldán-Alzate, A., and Johnson, K. M. (2021). Enhancement of cerebrovascular 4D flow MRI velocity fields using machine learning and computational fluid dynamics simulation data. *Sci. Rep*. 11, 1–11. doi: 10.1038/s41598-021-89636-z

Sánchez, I., and Vilaplana, V. (2018). "Brain MRI super-resolution using 3D generative adversarial networks," in *Proceedings of MIDL*.

Santelli, C., Loecher, M., Busch, J., Wieben, O., Schaeffter, T., and Kozerke, S. (2016). Accelerating 4D flow MRI by exploiting vector field divergence regularization. *Magn. Reson. Med*. 75, 115–125. doi: 10.1002/mrm.25563

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., et al. (2016). "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of CVPR*, 1874–1883.

Stalder, A. F., Russe, M., Frydrychowicz, A., Bock, J., Hennig, J., and Markl, M. (2008). Quantitative 2D and 3D phase contrast MRI: optimized analysis of blood flow and vessel wall parameters. *Magn. Reson. Med*. 60, 1218–1231. doi: 10.1002/mrm.21778

Stankovic, Z., Allen, B. D., Garcia, J., Jarvis, K. B., and Markl, M. (2014). 4D flow imaging with MRI. *Cardiovasc. Diagn. Ther*. 4, 173. doi: 10.3978/j.issn.2223-3652.2014.01.02

Tanno, R., Worrall, D. E., Ghosh, A., Kaden, E., Sotiropoulos, S. N., Criminisi, A., et al. (2017). "Bayesian image quality transfer with CNNs: exploring uncertainty in dMRI super-resolution," in *Proceedings of MICCAI* (Springer), 611–619.

Töger, J., Zahr, M. J., Aristokleous, N., Markenroth Bloch, K., Carlsson, M., and Persson, P.-O. (2020). Blood flow imaging by optimal matching of computational fluid dynamics to 4D-flow data. *Magn. Reson. Med*. 84, 2231–2245. doi: 10.1002/mrm.28269

Tsao, J., Boesiger, P., and Pruessmann, K. P. (2003). k-t BLAST and k-t SENSE: dynamic MRI with high frame rate exploiting spatiotemporal correlations. *Magn. Reson. Med*. 50, 1031–1042. doi: 10.1002/mrm.10611

Xie, Y., Franz, E., Chu, M., and Thuerey, N. (2018). tempogan: a temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Trans. Graphics* 37, 1–15. doi: 10.1145/3272127.3275078

Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., et al. (2018). Wide activation for efficient and accurate image super-resolution. *CoRR, abs/1808.08718*.

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. (2018). "Image super-resolution using very deep residual channel attention networks," in *Proceedings of ECCV*, 286–301.

clDice - a Novel Topology-Preserving Loss Function for Tubular Structure Segmentation

Suprosanna Shit, Johannes C. Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien P. W. Pluim, Ulrich Bauer & Bjoern H. Menze

Conference: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021

Synopsis: Accurate segmentation of tubular, network-like structures, such as vessels, neurons, or roads, is relevant to many fields of research. For such structures, the topology is their most important characteristic; particularly preserving connectedness: in the case of vascular networks, missing a connected vessel entirely alters the blood-flow dynamics. We introduce a novel similarity measure termed centerlineDice (short *clDice*), which is calculated on the intersection of the segmentation masks and their (morphological) skeleta. We theoretically prove that *clDice* guarantees topology preservation up to homotopy equivalence for binary 2D and 3D segmentation. Extending this, we propose a computationally efficient, differentiable loss function (*soft-clDice*) for training arbitrary neural segmentation networks. We benchmark the *soft-clDice* loss on five public datasets, including vessels, roads and neurons (2D and 3D). Training on *soft-clDice* leads to segmentation with more accurate connectivity information, higher graph similarity, and better volumetric scores.

Contributions of thesis author: Envisioned clDice as a loss function, developed a differentiable algorithm of soft-clDice loss, conceptualized theoretical support of clDice for topology preservation, executed 3D computational experiments, and carried out a leading role in planning the project and in organizing and writing the manuscript.

clDice - a Novel Topology-Preserving Loss Function for Tubular Structure Segmentation

Suprosanna Shit [*1] Johannes C. Paetzold [*1] Anjany Sekuboyina[1] Ivan Ezhov[1]
Alexander Unger[1] Andrey Zhylka[2] Josien P. W. Pluim[2] Ulrich Bauer[1] Bjoern H. Menze[1]
[1]Technical University of Munich [2] Eindhoven University of Technology

Abstract

Accurate segmentation of tubular, network-like structures, such as vessels, neurons, or roads, is relevant to many fields of research. For such structures, the topology is their most important characteristic; particularly preserving connectedness: in the case of vascular networks, missing a connected vessel entirely alters the blood-flow dynamics. We introduce a novel similarity measure termed centerlineDice (short clDice), which is calculated on the intersection of the segmentation masks and their (morphological) skeleta. We theoretically prove that clDice guarantees topology preservation up to homotopy equivalence for binary 2D and 3D segmentation. Extending this, we propose a computationally efficient, differentiable loss function (soft-clDice) for training arbitrary neural segmentation networks. We benchmark the soft-clDice loss on five public datasets, including vessels, roads and neurons (2D and 3D). Training on soft-clDice leads to segmentation with more accurate connectivity information, higher graph similarity, and better volumetric scores.

1. Introduction

Segmentation of *tubular* and *curvilinear* structures is an essential problem in numerous domains, such as clinical and biological applications (blood vessel and neuron segmentation from microscopic, optoacoustic, or radiology images), remote sensing applications (road network segmentation from satellite images) and industrial quality control, etc. In the aforementioned domains, a topologically accurate segmentation is necessary to guarantee error-free downstream tasks, such as computational hemodynamics, route planning, Alzheimer's disease prediction [18], or stroke modeling [20]. When optimizing computational algorithms for segmenting curvilinear structures, the two most commonly used categories of quantitative performance measures for evaluating segmentation accuracy of *tubular* structures, are 1) overlap based measures such as Dice, precision, recall, and Jaccard index; and 2) volumetric distance measures such as the Hausdorff and Mahalanobis distance [21, 40, 36, 16].

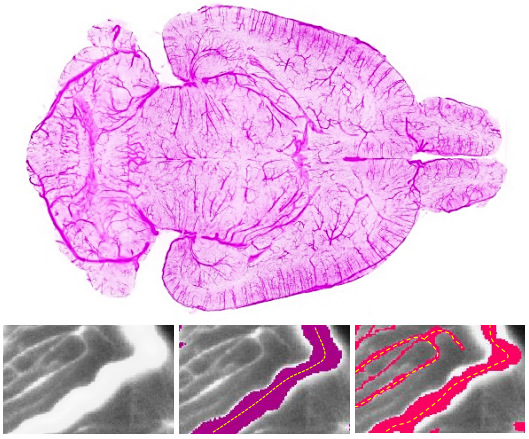

Figure 1. **Motivation:** The figure shows a 3D rendering of a complex, whole brain vascular dataset [48], where an exemplary 2D slice of the data is chosen and segmented by two different models, see purple (middle) and red (right), respectively. The two segmentation results achieve identical quality in terms of the traditional Dice score. Note that the purple segmentation does not capture the small vessels while segmenting the large vessel very accurately; on the other side, the red segmentation captures all vessels in the image while being less accurate on the radius of the large vessel. Skeleta are drawn in yellow. From a topology or network perspective, the red segmentation is evidently preferred.

However, in most segmentation problems, where the object of interest is 1) locally a *tubular* structure and 2) globally forms a *network*, the most important characteristic is the connectivity of the global network topology. Note that *network* in this context implies a physically connected structure, such as a vessel network, a road network, etc., which is also the primary structure of interest for the given

[*]The authors contributed equally to the work

image data. As an example, one can refer to brain vasculature analysis, where a missed vessel segment in the segmentation mask can pathologically be interpreted as a stroke or may lead to dramatic changes in a global simulation of blood flow. On the other hand, limited over- or under-segmentation of vessel radius can be tolerated, because it does not affect clinical diagnosis.

For evaluating segmentations in such tubular-network structures, traditional volume-based performance indices are sub-optimal. For example, Dice and Jaccard rely on the average voxel-wise hit or miss prediction [46]. In a task like network-topology extraction, a spatially contiguous sequence of correct voxel prediction is more meaningful than a spurious correct prediction. This ambiguity is relevant for objects of interest, which are of the same thickness as the resolution of the signal. For them, it is evident that a single-voxel shift in the prediction can change the topology of the whole network. Further, a globally averaged metric does not equally weight tubular-structures with large, medium, and small radii (cf. Fig 1). In real vessel datasets, where vessels of wide radius ranges exist, e.g. 30 μm for arterioles and 5 μm for capillaries [48, 9], training on a globally averaged loss induces a strong bias towards the volumetric segmentation of large vessels. Both scenarios are pronounced in imaging modalities, such as fluorescence microscopy [48, 58] and optoacoustics, which focus on mapping small capillary structures.

To this end, we are interested in a topology-aware image segmentation, eventually enabling a correct network extraction. Therefore, we ask the following research questions:

Q1. What is a good pixelwise measure to benchmark segmentation algorithms for **tubular**, and related linear and curvilinear structure segmentation while guaranteeing the preservation of the **network-topology**?

Q2. Can we use this *improved measure* as a loss function for neural networks, which is a void in existing literature?

1.1. Related Literature

Achieving topology preservation can be crucial to obtain meaningful segmentation, particularly for elongated and connected shapes, e.g. vascular structures or roads. However, analyzing preservation of topology while simplifying geometries is a difficult analytical and computational problem [11, 10].

For binary geometries, various algorithms based on thinning and medial surfaces have been proven to be topology-preserving according to varying definitions of topology [23, 25, 26, 35]. For non-binary geometries, existing methods applied topology and connectivity constraints onto variational and Markov random field-based methods: tree shape priors for vessel segmentation [44], graph representation priors to natural images [2], higher-order cliques which connect superpixels [53] and adversarial learning for road segmentation [51], integer programming to general curvilinear structures [49], and proposed a tree-structured convolutional gated recurrent unit [22], morphological optimization [14], among others [3, 15, 32, 31, 33, 37, 41, 52, 57, 56]. Further, topological priors of containment were applied to histology scans [5], a 3D CNN with graph refinement was used to improve airway connectivity [19], and recently, Mosinska et al. trained networks which perform segmentation and path classification simultaneously [30]. Another approach enables the predefinition of Betti numbers and enforces them on the training[8].

The aforementioned literature has advanced the communities understanding of topology-preservation, but critically, they do not possess end-to-end loss functions that optimize topology-preservation. In this context, the literature remains sparse. Recently, Mosinska et al. suggested that pixel-wise loss-functions are unsuitable and used selected filter responses from a VGG19 network as an additional penalty [29]. Nonetheless, their approach does not prove topology preservation. Importantly, Hu et al. proposed the first continuous-valued loss function based on the Betti number and persistent homology [17]. However, this method is based on matching critical points, which, according to the authors makes the training very expensive and error-prone for real image-sized patches [17]. While this is already limiting for a translation to large real world data set, we find that none of these approaches have been extended to three dimensional (3D) data.

1.2. Our Contributions

The objective of this paper is to identify an efficient, general, and intuitive loss function that enables topology preservation while segmenting tubular objects. We introduce a novel connectivity-aware similarity measure named *clDice* for benchmarking tubular-segmentation algorithms. Importantly, we provide theoretical guarantees for the topological correctness of the *clDice* for binary 2D and 3D segmentation. As a consequence of its formulation based on morphological skeletons, our measure pronounces the network's topology instead of equally weighting every voxel. Using a differentiable soft-skeletonization, we show that the *clDice* measure can be used to train neural networks. We show experimental results for various 2D and 3D network segmentation tasks to demonstrate the practical applicability of our proposed similarity measure and loss function.

2. Let's Emphasize *Connectivity*

We propose a novel connectivity-preserving metric to evaluate tubular and linear structure segmentation based on intersecting skeletons with masks. We call this metric *centerlineDice* or ***clDice***.

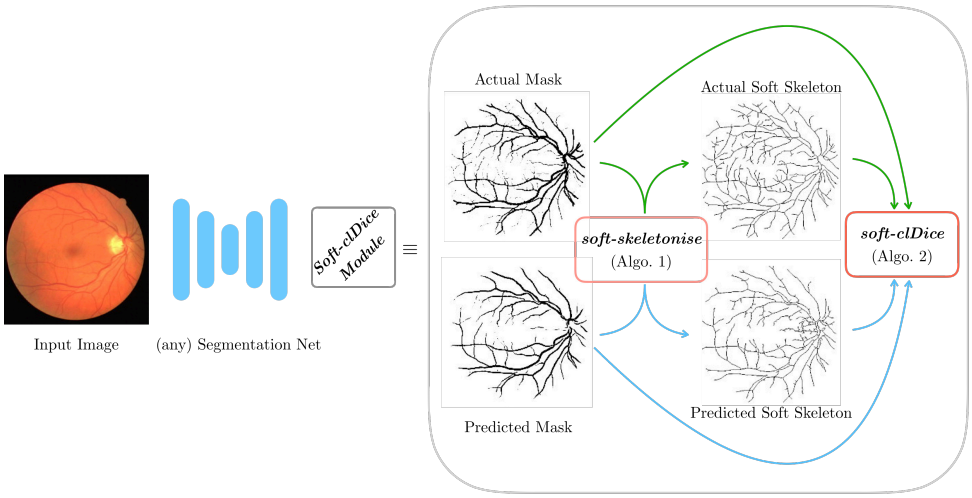

Figure 2. **Schematic overview of our proposed method:** Our proposed *clDice* loss can be applied to any arbitrary segmentation network. The soft-skeletonization can be easily implemented using pooling functions from any standard deep-learning toolbox.

We consider two binary masks: the ground truth mask (V_L) and the predicted segmentation masks (V_P). First, the skeletons S_P and S_L are extracted from V_P and V_L respectively. Subsequently, we compute the fraction of S_P that lies within V_L, which we call *Topology Precision* or $\mathrm{Tprec}(S_P, V_L)$, and vice-a-versa we obtain *Topology Sensitivity* or $\mathrm{Tsens}(S_L, V_P)$ as defined bellow;

$$\mathrm{Tprec}(S_P, V_L) = \frac{|S_P \cap V_L|}{|S_P|}; \quad \mathrm{Tsens}(S_L, V_P) = \frac{|S_L \cap V_P|}{|S_L|} \tag{1}$$

We observe that the measure $\mathrm{Tprec}(S_P, V_L)$ is susceptible to false positives in the prediction while the measure $\mathrm{Tsens}(S_L, V_P)$ is susceptible to false negatives. This explains our rationale behind referring to the $\mathrm{Tprec}(S_P, V_L)$ as topology's precision and to the $\mathrm{Tsens}(S_L, V_P)$ as its sensitivity. Since we want to maximize both precision and sensitivity (recall), we define *clDice* to be the harmonic mean (also known as F1 or Dice) of both the measures:

$$\mathrm{clDice}(V_P, V_L) = 2 \times \frac{\mathrm{Tprec}(S_P, V_L) \times \mathrm{Tsens}(S_L, V_P)}{\mathrm{Tprec}(S_P, V_L) + \mathrm{Tsens}(S_L, V_P)} \tag{2}$$

Note that our *clDice* formulation is not defined for $\mathrm{Tprec} = 0$ and $\mathrm{Tsens} = 0$, but can easily be extended continuously with the value 0.

3. Topological Guarantees for clDice

The following section provides general theoretical guarantees for the preservation of topological properties achieved by optimizing *clDice* under mild conditions on the input. Roughly, these conditions state that the object of interest is embedded in S^3 in a non-knotted way, as is typically the case for blood vessel and road structures.

Specifically, we assume that both ground truth and prediction *admit foreground and background skeleta*, which means that both foreground and background are homotopy-equivalent to topological graphs, which we assume to be embedded as *skeleta*. Here, the voxel grid is considered as a cubical complex, consisting of elementary cubes of dimensions 0, 1, 2, and 3. This is a special case of a *cell complex* (specifically, a *CW complex*), which is a space constructed inductively, starting with isolated points (0-cells), and gluing a collection of topological balls of dimension k (called k-*cells*) along their boundary spheres to a $k-1$-dimensional complex. The voxel grid, seen as a cell complex in this sense, can be completed to an ambient complex that is homeomorphic to the 3-sphere S^3 by attaching a single exterior cell to the boundary. In order to consider foreground and background of a binary image as complementary subspaces, the foreground is now assumed to be the union of closed unit cubes in the voxel grid, corresponding to voxels with value 1; and the background is the complement in the ambient complex. This convention is commonly used in digital topology [24, 23]. The assumption on the background can then be replaced by a convenient equivalent condition, stating that the foreground is also homotopy equivalent to a subcomplex obtained from the ambient complex by only removing 3-cells and 2-cells. Such a subcomplex is then clearly homotopy-equivalent to the complement

of a 1-complex.

We will now observe that the above assumptions imply that the foreground and the background are connected and have a free fundamental group and vanishing higher fundamental groups. In particular, the homotopy type is already determined by the first Betti number [1]; moreover, a map inducing an isomorphism in homology is already a homotopy equivalence. To see this, first note that both foreground and background are assumed to have the homology of a graph, in particular, homology is trivial in degree 2. By Alexander duality [1], then, both foreground and background have trivial reduced cohomology in degree 0, meaning that they are connected. This implies that both have a free fundamental group (as any connected graph) and vanishing higher homotopy groups. In particular, since homology in degree 1 is the Abelianization of the fundamental group, these two groups are isomorphic. This in turn implies that in our setting a map that induces isomorphisms in homology already induces isomorphisms between all homotopy groups. By Whitehead's theorem [54], such a map is then a homotopy equivalence.

The following theorem shows that under our assumptions on the images admitting foreground and background skeleta, the existence of certain nested inclusions already implies the homotopy-equivalence of foreground and background, which we refer to as *topology preservation*.

Theorem 1. *Let $L_A \subseteq A \subseteq K_A$ and $L_B \subseteq B \subseteq K_B$ be connected subcomplexes of some cell complex. Assume that the above inclusions are homotopy equivalences. If the subcomplexes also are related by inclusions $L_A \subseteq B \subseteq K_A$ and $L_B \subseteq A \subseteq K_B$, then these inclusions must be homotopy equivalences as well. In particular, A and B are homotopy-equivalent.*

Proof. An inclusion of cell complexes map is a homotopy equivalence if and only if it induces isomorphisms on all homotopy groups. Since the inclusion $L_A \subseteq B \subseteq K_A$ induces an isomorphism, the inclusion $L_A \subseteq B$ induces a left-inverse, and since $B \subseteq K_B$ induces an isomorphism, the inclusion $L_A \subseteq K_B$ also induces a left-inverse. At the same time, since the inclusion $L_B \subseteq A \subseteq K_B$ induces an isomorphism, the inclusion $A \subseteq K_B$ induces a left-inverse, and since $L_A \subseteq A$ induces an isomorphism, the inclusion $L_A \subseteq K_B$ also induces a right-inverse. Together, this implies that the inclusion $L_A \subseteq K_B$ induces an isomorphism.

Together with the isomorphisms induced by $L_A \subseteq A$ and $B \subseteq K_B$, we obtain isomorphisms induced by $L_A \subseteq B$ and by $A \subseteq K_B$, which compose to an isomorphism between the homotopy groups of A and B. □

[1]Betti numbers: β_0 represents the number of distinct *connected-components*, β_1 represents the number of *circular holes*, and β_2 represents the number of *cavities*, for depictions see Supplementary material

Corollary 1.1. *Let V_L and V_P be two binary masks admitting foreground and background skeleta, such that the foreground skeleton of V_L is included in the foreground of V_P and vice versa, and similarly for the background. Then the foregrounds of V_L and V_P are homotopy equivalent, and the same is true for their backgrounds.*

Note that the inclusion condition in this corollary is satisfied if and only if *clDice* evaluates to 1 on both foreground and background of (V_L, V_P).

This proof lays the ground for a general interpretation of *clDice* as a topology preserving metric. Additionally, we provide an elaborate explanation of *clDice* topological properties, using concepts of applied digital topology in the theory section of the Supplementary material [24, 23].

4. Training Neural Networks with *clDice*

In the previous section we provided general theoretic guarantees how *clDice* has topology preserving properties. The following chapter shows how we applied our theory to efficiently train topology preserving networks using the *clDice* formulation. [2]

4.1. *Soft-clDice* using *Soft-skeletonization*:

Extracting accurate skeletons is essential to our method. For this task, a multitude of approaches has been proposed. However, most of them are not fully differentiable and therefore unsuited to be used in a loss function. Popular approaches use the Euclidean distance transform or utilize repeated morphological thinning. Euclidean distance transform has been used on multiple occasions [42, 55], but remains a discrete operation and, to the best of our knowledge, an end-to-end differentiable approximation remains to be developed, preventing the use in a loss function for training neural networks. On the contrary, morphological thinning is a sequence of dilation and erosion operations [c.f. Fig. 3].

Importantly, thinning using morphological operations (skeletonization) on curvilinear structures can be topology-preserving [35]. Min- and max filters are commonly used as the grey-scale alternative of morphological dilation and erosion. Motivated by this, we propose 'soft-skeletonization', where an iterative min- and max-pooling is applied as a proxy for morphological erosion and dilation. The Algorithm 1 describes the iterative processes involved in its computation. The hyper-parameter k involved in its computation represents the iterations and has to be greater than or equal to the maximum observed radius. In our experiments, this parameter depends on the dataset. For example, it is $k = 5...25$ in our experiments, matching the pixel radius of the largest observed tubular structures. Choosing a larger k does not reduce performance but increases computation

[2]https://github.com/jocpae/clDice

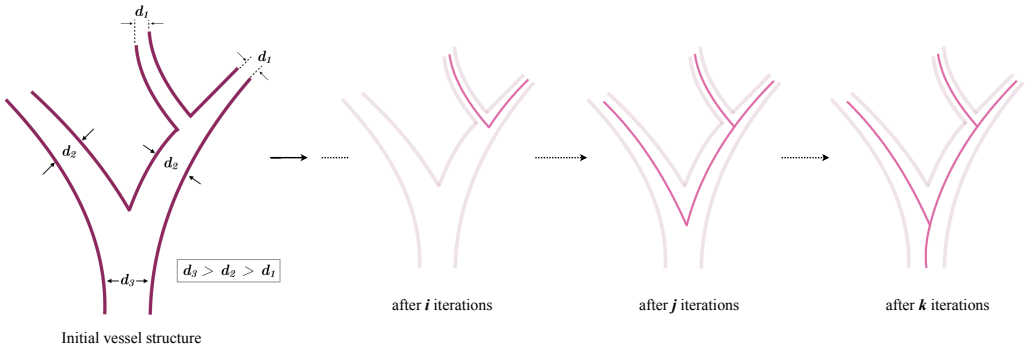

Figure 3. Based on the initial vessel structure (purple), sequential bagging of skeleton voxels (red) via iterative skeletonization leads to a complete skeletonization, where d denotes the diameter and $k > j > i$ iterations.

Algorithm 1: *soft-skeleton*

Input: I, k

$I' \leftarrow \mathit{maxpool}(\mathit{minpool}(I))$

$S \leftarrow \mathit{ReLU}(I - I')$

for $i \leftarrow 0$ **to** k **do**

$I \leftarrow \mathit{minpool}(I)$

$I' \leftarrow \mathit{maxpool}(\mathit{minpool}(I))$

$S \leftarrow S + (1 - S) \circ \mathit{ReLU}(I - I')$

end

Output: S

Algorithm 2: *soft-clDice*

Input: V_P, V_L

$S_P \leftarrow \mathit{soft\text{-}skeleton}(V_P)$

$S_L \leftarrow \mathit{soft\text{-}skeleton}(V_L)$

$\mathit{Tprec}(S_P, V_L) \leftarrow \frac{|S_P \circ V_L| + \epsilon}{|S_P| + \epsilon}$

$\mathit{Tsens}(S_L, V_P) \leftarrow \frac{|S_L \circ V_P| + \epsilon}{|S_L| + \epsilon}$

$\mathit{clDice} \leftarrow 2 \times \frac{\mathit{Tprec}(S_P, V_L) \times \mathit{Tsens}(S_L, V_P)}{\mathit{Tprec}(S_P, V_L) + \mathit{Tsens}(S_L, V_P)}$

Output: $clDice$

Figure 4. **Algorithm 1** calculates the proposed *soft-skeleton*, here I is the mask to be *soft-skeletonized* and k is the number of iterations for skeletonization. **Algorithm 2**, calculates the *soft-clDice* loss, where V_P is a real-valued probabilistic prediction from a segmentation network and V_L is the true mask. We denote Hadamard product using $\circ$.

time. On the other hand, a too low k leads to incomplete skeletonization.

In Figure 3, the successive steps of our skeletonization are intuitively represented. In the early iterations, the structures with a small radius are skeletonized and preserved until the later iterations when the thicker structures become skeletonized. This enables the extraction of a parameter-free, morphologically motivated soft-skeleton. The aforementioned soft-skeletonization enables us to use *clDice* as a fully differentiable, real-valued, optimizable measure. The Algorithm 2 describes its implementation. We refer to this as the *soft-clDice*.

For a single connected foreground component and in the absence of knots, the homotopy type is specified by the number of linked loops. Hence, if the reference and the predicted volumes are not homotopy equivalent, they do not have pairwise linked loops. To include these missing loops or exclude the extra loops, one has to add or discard deformation retracted skeleta of the solid foreground. This implies adding *new correctly predicted voxels*. In contrast to other volumetric losses such as Dice, cross-entropy, etc., *clDice* only considers the deformation-retracted graphs of the solid foreground structure. Thus, we claim that *clDice* requires the least amount of *new correctly predicted voxels* to guarantee the homotopy equivalence. Along these lines, Dice or cross-entropy can only guarantee homotopy equivalence if every single voxel is segmented correctly. On the other hand, *clDice* can guarantee homotopy equivalence for a broader combinations of connected-voxels. Intuitively, this is a very much desirable property as it makes *clDice* robust towards outliers and noisy segmentation labels.

4.2. Cost Function

Since our objective here is to preserve topology while achieving accurate segmentations, and not to learn skeleta, we combine our proposed *soft-clDice* with *soft-Dice* in the following manner:

$$\mathcal{L}_c = (1 - \alpha)(1 - \mathit{soft}\textbf{\textit{Dice}}) + \alpha(1 - \mathit{soft}\textbf{\textit{clDice}}) \quad (3)$$

where $\alpha \in [0, 0.5]$. In stark contrast to previous works, where segmentation and centerline prediction has been learned jointly as multi-task learning [50, 47], we are not interested in learning the centerline. We are interested in

learning a topology-preserving segmentation. Therefore, we restrict our experimental choice of alpha to $\alpha \in [0, 0.5]$. We test *clDice* on two state-of-the-art network architectures: i) a 2D and 3D U-Net[38, 6], and ii) a 2D and 3D fully connected networks (FCN) [47, 13]. As baselines, we use the same architectures trained using *soft-Dice* [27, 45].

4.3. Adaption for Highly Imbalanced Data

Our theory (Section 3), describes a two-class problem where *clDice* should be computed on both the foreground and the background channels. In our experiments, we show that for complex and highly imbalanced dataset it is sufficient to calculate the **clDice** loss on the underrepresented foreground class. We attribute this to the distinct properties of tubularness, sparsity of foreground and the lack of cavities (Betti number 2) in our data. An intuitive interpretation how these assumptions are valid in terms of digital topology can be found in the supplementary material.

5. Experiments

5.1. Datasets

We employ five public datasets for validating *clDice* and *soft-clDice* as a measure and an objective function, respectively. In 2D, we evaluate on the DRIVE retina dataset [43], the Massachusetts Roads dataset [28] and the CREMI neuron dataset [12]. In 3D, a synthetic vessel dataset with an added Gaussian noise term [39] and the Vessap dataset of multi-channel volumetric scans of brain vessels is used [48, 34]. For the Vessap dataset we train different models for one and two input channels. For all of the datasets, we perform three fold cross-validation and test on held-out, large, and highly-variant test sets. Details concerning the experimental setup can be found in the supplementary material.

5.2. Evaluation Metrics

We compare the performance of various experimental setups using three types of metrics: volumetric, topology-based, and graph-based.

1. Volumetric: We compute volumetric scores such as Dice coefficient, Accuracy, and the proposed *clDice*.
2. Topology-based: We calculate the mean of absolute Betti Errors for the Betti Numbers β_0 and β_1 and the mean absolute error of Euler characteristic, $\chi = V - E + F$, where $V, E,$ and F denotes number of vertices, edges, and faces.
3. Graph-based: we extract random patch-wise graphs for the 2D/3D images. We uniformly sample fixed number of points from the graph and compute the StreetmoverDistance (SMD) [4]. SMD captures a Wasserstein distance between two graphs. Additionally we compute the F1 score of junction-based metric [7].

5.3. Results and Discussion

We trained two segmentation architectures, a U-Net and an FCN, for the various loss functions in our experimental setup. As a baseline, we trained the networks using *soft-dice* and compared it with the ones trained using the proposed loss (Eq. 3), by varying α from (0.1 to 0.5).

Quantitative: We observe that including *soft-clDice* in any proportion ($\alpha > 0$) leads to improved topological, volumetric and graph similarity for all 2D and 3D datasets, see Table 1. We conclude that α can be interpreted as a hyper parameter which can be tuned *per-dataset*. Intuitively, increasing the α improves the *clDice* measure for most experiments. Most often, *clDice* is high or highest when the graph and topology based measures are high or highest, particularly the β_1 Error, Streetmover distance and Opt-J F1 score; quantitatively indicating that topological properties are indeed represented in the *clDice* measure.

In spite of not optimizing for a high *soft-clDice* on the background class, all of our networks converge to superior segmentation results. This not only reinforces our assumptions on dataset-specific necessary conditions but also validates the practical applicability of our loss. Our findings hold for the different network architectures, for 2D or 3D, and for tubular or curvilinear structures, strongly indicating its generalizability to analogous binary segmentation tasks.

Observe that CREMI and the synthetic vessel dataset (see Supplementary material) appear to have the smallest increase in scores over the baseline. We attribute this to them being the least complex datasets in the collection, with CREMI having an almost uniform thickness of radii and the synthetic data having a high signal-to-noise ratio and insignificant illumination variation. More importantly, we observe larger improvements for all measures in case of the more complex Vessap and Roads data see Figure 5. In direct comparison to performance measures reported in two recent publications by Hu et al. and Mosinska et al. [17, 29], we find that our approach is on par or better in terms of Accuracy and Betti Error for the Roads and CREMI dataset. It is important to note that we used a smaller subset of training data for the Road dataset compared to both while using the same test set.

Hu et al. reported a Betti error for the DRIVE data, which exceeds ours; however, it is important to consider that their approach explicitly minimizes the mismatch of the persistence diagram, which has significantly higher computational complexity during training, see the section below. We find that our proposed loss performs superior to the baseline in almost every scenario. The improvement appears to be pronounced when evaluating the highly relevant graph and topology based measures, including the recently

Table 1. Quantitative experimental results for the Massachusetts road dataset (Roads), the CREMI dataset, the DRIVE retina dataset and the Vessap dataset (3D). Bold numbers indicate the best performance. The performance according to the *clDice* measure is highlighted in rose. For all experiments we observe that using *soft-clDice* in $\mathcal{L}_c$ results in improved scores compared to *soft-Dice*. This improvement holds for almost $\alpha > 0$; α can be interpreted as a dataset specific hyper-parameter.

Dataset	Network	Loss	Dice	Accuracy	*clDice*	β_0 Error	β_1 Error	SMD [4]	χ_{error}	Opt-J F1 [7]
Roads	FCN	*soft-dice*	64.84	95.16	70.79	1.474	1.408	0.1216	2.634	0.766
		$\mathcal{L}_c, \alpha = 0.1$	66.52	95.70	74.80	0.987	1.227	0.1002	2.625	0.768
		$\mathcal{L}_c, \alpha = 0.2$	**67.42**	**95.80**	76.25	**0.920**	1.280	**0.0954**	2.526	0.770
		$\mathcal{L}_c, \alpha = 0.3$	65.90	95.35	74.86	0.974	1.197	0.1003	2.448	0.775
		$\mathcal{L}_c, \alpha = 0.4$	67.18	95.46	**76.92**	0.934	**1.092**	0.0991	**2.183**	**0.803**
		$\mathcal{L}_c, \alpha = 0.5$	65.77	95.09	75.22	0.947	1.184	0.0991	2.361	0.782
	U-NET	*soft-dice*	76.23	96.75	86.83	0.491	1.256	0.0589	1.120	0.881
		$\mathcal{L}_c, \alpha = 0.1$	**76.66**	**96.77**	87.35	0.359	**0.938**	0.0457	0.980	0.878
		$\mathcal{L}_c, \alpha = 0.2$	76.25	96.76	87.29	**0.312**	1.031	**0.0415**	0.865	0.900
		$\mathcal{L}_c, \alpha = 0.3$	74.85	96.57	86.10	0.322	1.062	0.0504	0.827	0.913
		$\mathcal{L}_c, \alpha = 0.4$	75.38	96.60	86.16	0.344	1.016	0.0483	**0.755**	**0.916**
		$\mathcal{L}_c, \alpha = 0.5$	76.45	96.64	**88.17**	0.375	0.953	0.0527	1.080	0.894
	Mosinska et al.	[29, 17]	-	97.54	-	-	2.781	-	-	-
	Hu et al.	[17]	-	97.28	-	-	1.275	-	-	-
CREMI	U-NET	*soft-dice*	91.54	97.11	95.86	0.259	0.657	0.0461	1.087	0.904
		$\mathcal{L}_c, \alpha = 0.1$	91.76	**97.21**	96.05	0.222	0.556	**0.0395**	1.000	0.900
		$\mathcal{L}_c, \alpha = 0.2$	91.66	97.15	96.01	0.231	0.630	0.0419	0.991	0.902
		$\mathcal{L}_c, \alpha = 0.3$	**91.78**	97.18	**96.21**	**0.204**	**0.537**	0.0437	**0.919**	**0.913**
		$\mathcal{L}_c, \alpha = 0.4$	91.56	97.12	96.09	0.250	0.630	0.0444	0.995	0.902
		$\mathcal{L}_c, \alpha = 0.5$	91.66	97.16	96.16	0.231	0.620	0.0455	0.991	0.907
	Mosinska et al.	[29, 17]	82.30	94.67	-	-	1.973	-	-	-
	Hu et al.	[17]	-	94.56	-	-	1.113	-	-	-
DRIVE retina	FCN	*soft-Dice*	78.23	96.27	78.02	2.187	1.860	0.0429	3.275	0.773
		$\mathcal{L}_c, \alpha = 0.1$	78.36	96.25	79.02	2.100	1.610	0.0393	3.203	0.777
		$\mathcal{L}_c, \alpha = 0.2$	**78.75**	96.29	80.22	1.892	1.382	0.0383	2.895	0.793
		$\mathcal{L}_c, \alpha = 0.3$	78.29	96.20	80.28	1.888	**1.332**	**0.0318**	2.918	**0.798**
		$\mathcal{L}_c, \alpha = 0.4$	78.00	96.11	80.43	2.036	1.602	0.0423	3.141	0.764
		$\mathcal{L}_c, \alpha = 0.5$	77.76	96.04	**80.95**	**1.836**	1.408	0.0394	**2.848**	0.794
	U-Net	*soft-Dice*	74.25	95.63	75.71	1.745	1.455	0.0649	2.997	0.760
		$\mathcal{L}_c, \alpha = 0.5$	**75.21**	**95.82**	**76.86**	**1.538**	**1.389**	**0.0586**	**2.737**	**0.767**
	Mosinska et al.	[29, 17]	-	95.43	-	-	2.784	-	-	-
	Hu et al.	[17]	-	95.21	-	-	1.076	-	-	-
Vessap data	FCN, 1 ch	*soft-dice*	85.21	**96.03**	90.88	3.385	4.458	0.00459	5.850	0.862
		$\mathcal{L}_c, \alpha = 0.5$	**85.44**	95.91	**91.32**	**2.292**	**3.677**	**0.00417**	**5.620**	**0.864**
	FCN, 2 ch	*soft-dice*	85.31	95.82	90.10	2.833	4.771	0.00629	6.080	0.849
		$\mathcal{L}_c, \alpha = 0.1$	85.96	95.99	91.02	2.896	**4.156**	0.00447	5.980	0.860
		$\mathcal{L}_c, \alpha = 0.2$	**86.45**	**96.11**	91.22	2.656	4.385	0.00466	5.530	0.869
		$\mathcal{L}_c, \alpha = 0.3$	85.72	95.93	91.20	2.719	4.469	**0.00423**	5.470	0.866
		$\mathcal{L}_c, \alpha = 0.4$	85.65	95.95	**91.65**	2.719	4.469	**0.00423**	5.670	0.869
		$\mathcal{L}_c, \alpha = 0.5$	85.28	95.76	91.22	**2.615**	4.615	0.00433	**5.320**	**0.870**
	U-Net, 1 ch	*soft-dice*	87.46	96.35	91.18	3.094	5.042	0.00549	5.300	0.863
		$\mathcal{L}_c, \alpha = 0.5$	**87.82**	**96.52**	**93.03**	**2.656**	**4.615**	**0.00533**	**4.910**	**0.872**
	U-Net, 2 ch	*soft-dice*	87.98	96.56	90.16	2.344	4.323	0.00507	5.550	0.855
		$\mathcal{L}_c, \alpha = 0.1$	88.13	96.59	91.12	2.302	4.490	0.00465	5.180	**0.872**
		$\mathcal{L}_c, \alpha = 0.2$	87.96	96.74	92.52	2.208	**3.979**	0.00342	**4.830**	0.861
		$\mathcal{L}_c, \alpha = 0.3$	87.70	96.71	92.56	**2.115**	4.521	**0.00309**	5.260	0.858
		$\mathcal{L}_c, \alpha = 0.4$	**88.57**	**96.87**	**93.25**	2.281	4.302	0.00327	5.370	0.868
		$\mathcal{L}_c, \alpha = 0.5$	88.14	96.74	92.75	2.135	4.125	0.00328	5.390	0.864

introduced OPT-Junction F1 by Citraro et al. [7]. Our results are consistent across different network architectures, indicating that *soft-clDice* can be deployed to any network architecture.

Qualitative: In Figure 5, typical results for our datasets are depicted. Our networks trained on the proposed loss term recover connections, which were false negatives when trained with the soft-Dice loss. These missed connections appear to be particularly frequent in the complex road and DRIVE dataset. For the CREMI dataset, we observe these situations less frequently, which is in line with the very high quantitative scores on the CREMI data. Interestingly, in the real 3D vessel dataset, the soft-Dice loss oversegments vessels, leading to false positive connections. This is not the case when using our proposed loss function, which we attribute to its topology-preserving nature. Additional

qualitative results can be inspected in the supplementary.

Computational Efficiency: Naturally, inference times of CNNs with the same architecture but different training losses are identical. However, during training, our soft-skeleton algorithm requires $O(kn^2)$ complexity for an $n \times n$ 2D image where k is the number of iterations. As a comparison, [17] needs $O(c^2 mlog(m))$ (see [15]) complexity to compute the 1d persistent homology where d is the number of points with zero gradients in the prediction and m is the number of simplices. Roughly, c is proportional to n^2, and m is of $O(n^2)$ for a 2D Euclidean grid. Thus, the worst complexity of [17] is $O(n^6 log(n))$. Additionally, their approach requires an $O(clog(c))$ complexity to find an optimal matching of the birth-death pairs. We note that the total run-time overhead for soft-clDice compared to soft-Dice is marginal, i.e., for batch-size of 4 and 1024x1024 image resolution, the former takes 1.35s while the latter takes 1.24s on average ($<$10% increase) on an RTX-8000.

Future Work: Although our proposed soft-skeleton approximation works well in practice, a better differentiable skeletonization can only improve performance, which we reserve for future research. Any such skeletonization can be readily plugged into our approach. Furthermore, theoretical and experimental multi-class studies would sensibly extend our study.

6. Conclusive Remarks

We introduce *clDice*, a novel topology-preserving similarity measure for tubular structure segmentation. Importantly, we present a theoretical guarantee that *clDice* enforces topology preservation up to homotopy equivalence. Next, we use a differentiable version of the *clDice*, *soft-clDice*, in a loss function, to train state-of-the-art 2D and 3D neural networks. We use *clDice* to benchmark segmentation quality from a topology-preserving perspective along with multiple volumetric, topological, and graph-based measures. We find that training on *soft-clDice* leads to segmentations with more accurate connectivity information, better graph-similarity, better Euler characteristics, and improved Dice and Accuracy. Our *soft-clDice* is computationally efficient and can be readily deployed to any other deep learning-based segmentation tasks such as neuron segmentation in biomedical imaging, crack detection in industrial quality control, or remote sensing.

Acknowledgement: J. C. Paetzold. and S. Shit. are supported by the GCB and Translatum, TU Munich. S.Shit., A. Zhylka. and I. Ezhov. are supported by TRABIT (EU Grant: 765148). We thank Ali Ertuerk, Mihail I. Todorov, Nils Börner and Giles Tetteh.

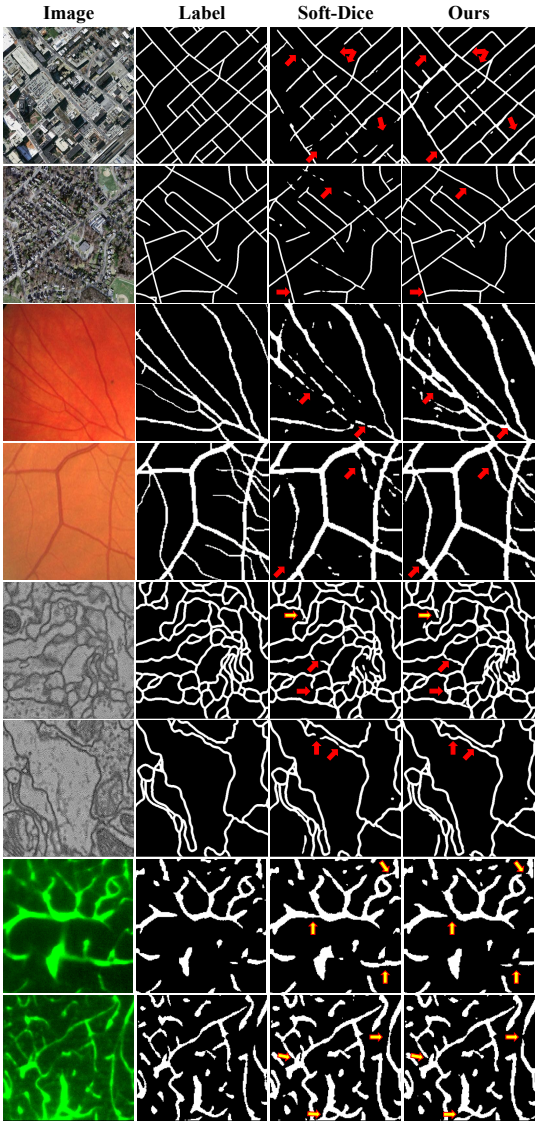

Figure 5. Qualitative results: from top to bottom we show two rows of results for: the Massachusetts road dataset, the DRIVE retina dataset, the CREMI neuron data and 2D slices from the 3D Vessap dataset. From left to right, the real image, the label, the prediction using soft-Dice and the U-Net predictions using $\mathcal{L}_c(\alpha = 0.5)$ are shown, respectively. The images indicate that *clDice* segments road, retina vessel connections and neuron connections which the soft-Dice loss misses, but also does not segment false-positive vessels in 3D. Some, but not all, missed connections are indicated with solid red arrows, false positives are indicated with red-yellow arrows. More qualitative results can be found in the Supplementary material.

References

[1] Pavel S Aleksandrov. *Combinatorial topology*, volume 1. Courier Corporation, 1998.

[2] Bjoern Andres et al. Probabilistic image segmentation with closedness constraints. In *ICCV*, pages 2611–2618. IEEE, 2011.

[3] Ricardo J Araújo, Jaime S Cardoso, and Hélder P Oliveira. A deep learning design for improving topology coherence in blood vessel segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 93–101. Springer, 2019.

[4] Davide Belli and Thomas Kipf. Image-conditioned graph generation for road network extraction. *arXiv preprint arXiv:1910.14388*, 2019.

[5] Aïcha BenTaieb and Ghassan Hamarneh. Topology aware fully convolutional networks for histology gland segmentation. In *MICCAI*, pages 460–468. Springer, 2016.

[6] Özgün Çiçek and Aothers. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *MICCAI*, pages 424–432. Springer, 2016.

[7] Leonardo Citraro, Mateusz Koziński, and Pascal Fua. Towards reliable evaluation of algorithms for road network reconstruction from aerial images. In *European Conference on Computer Vision*, pages 703–719. Springer, 2020.

[8] James Clough, Nicholas Byrne, Ilkay Oksuz, Veronika A Zimmer, Julia A Schnabel, and Andrew King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[9] Antonino Paolo Di Giovanna et al. Whole-brain vasculature reconstruction at the single capillary level. *Scientific reports*, 8(1):12573, 2018.

[10] Herbert Edelsbrunner et al. Topological persistence and simplification. In *FOCS*, pages 454–463. IEEE, 2000.

[11] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[12] Jan Funke, Fabian Tschopp, William Grisaitis, Arlo Sheridan, Chandan Singh, Stephan Saalfeld, and Srinivas C. Turaga. Large scale image segmentation with structured loss based deep learning for connectome reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1669–1680, Jul 2019.

[13] Stefan Gerl et al. A distance-based loss for smooth and continuous skin layer segmentation in optoacoustic images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 309–319. Springer, 2020.

[14] Shir Gur, Lior Wolf, Lior Golgher, and Pablo Blinder. Unsupervised microvascular image segmentation using an active contours mimicking neural network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10722–10731, 2019.

[15] Xiao Han et al. A topology preserving level set method for geometric deformable models. *IEEE TPAMI*, 25(6):755–768, 2003.

[16] Kai Hu et al. Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function. *Neurocomputing*, 309:179–191, 2018.

[17] Xiaoling Hu et al. Topology-preserving deep image segmentation. In *NeurIPS*, pages 5658–5669, 2019.

[18] Jesse M Hunter et al. Morphological and pathological evolution of the brain microcirculation in aging and Alzheimer's disease. *PloS one*, 7(5):e36893, 2012.

[19] Dakai Jin, Ziyue Xu, Adam P Harrison, Kevin George, and Daniel J Mollura. 3d convolutional neural networks with graph refinement for airway segmentation using incomplete data labels. In *International Workshop on Machine Learning in Medical Imaging*, pages 141–149. Springer, 2017.

[20] Anne Joutel et al. Cerebrovascular dysfunction and microcirculation rarefaction precede white matter lesions in a mouse genetic model of cerebral ischemic small vessel disease. *JCI*, 120(2):433–445, 2010.

[21] Cemil Kirbas and Francis Quek. A review of vessel extraction techniques and algorithms. *CSUR*, 36(2):81–121, 2004.

[22] Bin Kong, Xin Wang, Junjie Bai, Yi Lu, Feng Gao, Kunlin Cao, Jun Xia, Qi Song, and Youbing Yin. Learning tree-structured representation for 3d coronary artery segmentation. *Computerized Medical Imaging and Graphics*, 80:101688, 2020.

[23] T. Yung Kong. On topology preservation in 2-D and 3-D thinning. *International journal of pattern recognition and artificial intelligence*, 9(05):813–844, 1995.

[24] T Yung Kong and Azriel Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48(3):357–393, 1989.

[25] Ta-Chih Lee et al. Building skeleton models via 3-D medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, 1994.

[26] Cherng Min Ma. On topology preservation in 3D thinning. *CVGIP: Image understanding*, 59(3):328–339, 1994.

[27] Fausto Milletari et al. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, pages 565–571. IEEE, 2016.

[28] Volodymyr Mnih. *Machine Learning for Aerial Image Labeling*. PhD thesis, University of Toronto, 2013.

[29] Agata Mosinska et al. Beyond the pixel-wise loss for topology-aware delineation. In *CVPR*, pages 3136–3145, 2018.

[30] Agata Mosinska, Mateusz Koziński, and Pascal Fua. Joint segmentation and path classification of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(6):1515–1521, 2019.

[31] Fernando Navarro, Suprosanna Shit, et al. Shape-aware complementary-task learning for multi-organ segmentation. In *International Workshop on MLMI*, pages 620–627. Springer, 2019.

[32] Sebastian Nowozin and Christoph H Lampert. Global connectivity potentials for random field models. In *CVPR*, pages 818–825. IEEE, 2009.

[33] Martin Ralf Oswald et al. Generalized connectivity constraints for spatio-temporal 3D reconstruction. In *ECCV*, pages 32–46. Springer, 2014.

[34] Johannes C Paetzold, Oliver Schoppe, et al. Transfer learning from synthetic data reduces need for labels to segment brain vasculature and neural pathways in 3d. In *International Conference on Medical Imaging with Deep Learning–Extended Abstract Track*, 2019.

[35] Kálmán Palágyi. A 3-subiteration 3D thinning algorithm for extracting medial surfaces. *Pattern Recognition Letters*, 23(6):663–675, 2002.

[36] Renzo Phellan et al. Vascular segmentation in TOF MRA images of the brain using a deep convolutional neural network. In *MICCAI Workshop*, pages 39–46. Springer, 2017.

[37] Markus Rempfler et al. Efficient algorithms for moral lineage tracing. In *ICCV*, pages 4695–4704, 2017.

[38] Olaf Ronneberger et al. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015.

[39] Matthias Schneider et al. Tissue metabolism driven arterial tree generation. *Med Image Anal.*, 16(7):1397–1414, 2012.

[40] Matthias Schneider et al. Joint 3-D vessel segmentation and centerline extraction using oblique Hough forests with steerable filters. *Med Image Anal.*, 19(1):220–249, 2015.

[41] Florent Ségonne. Active contours under topology control—genus preserving level sets. *International Journal of Computer Vision*, 79(2):107–117, 2008.

[42] Frank Y Shih and Christopher C Pu. A skeletonization algorithm by maxima tracking on euclidean distance transform. *Pattern Recognition*, 28(3):331–341, 1995.

[43] Joes Staal, Michael D Abràmoff, Meindert Niemeijer, Max A Viergever, and Bram Van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging*, 23(4):501–509, 2004.

[44] Jan Stuhmer et al. Tree shape priors with connectivity constraints using convex relaxation on general graphs. In *ICCV*, pages 2336–2343, 2013.

[45] Carole H Sudre et al. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *MICCAI Workshop*, pages 240–248. Springer, 2017.

[46] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, 15(1):29, 2015.

[47] Giles Tetteh et al. Deepvesselnet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes. *arXiv preprint arXiv:1803.09340*, 2018.

[48] Mihail Ivilinov Todorov, Johannes C. Paetzold, et al. Automated analysis of whole brain vasculature using machine learning. *bioRxiv*, page 613257, 2019.

[49] Engin Türetken et al. Reconstructing curvilinear networks using path classifiers and integer programming. *IEEE TPAMI*, 38(12):2515–2530, 2016.

[50] Fatmatülzehra Uslu and Anil Anthony Bharath. A multi-task network to detect junctions in retinal vasculature. In *MICCAI*, pages 92–100. Springer, 2018.

[51] Subeesh Vasu, Mateusz Kozinski, Leonardo Citraro, and Pascal Fua. Topoal: An adversarial learning approach for topology-aware road segmentation. *arXiv preprint arXiv:2007.09084*, 2020.

[52] Sara Vicente et al. Graph cut based image segmentation with connectivity priors. In *CVPR*, pages 1–8. IEEE, 2008.

[53] Jan D Wegner et al. A higher-order CRF model for road network extraction. In *CVPR*, pages 1698–1705. IEEE, 2013.

[54] John HC Whitehead. Combinatorial homotopy. i. *Bulletin of the American Mathematical Society*, 55(3):213–245, 1949.

[55] Mark W Wright et al. Skeletonization using an extended euclidean distance transform. *Image and Vision Computing*, 13(5):367–375, 1995.

[56] Aaron Wu, Ziyue Xu, Mingchen Gao, Mario Buty, and Daniel J Mollura. Deep vessel tracking: A generalized probabilistic approach via deep learning. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 1363–1367. IEEE, 2016.

[57] Yun Zeng et al. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n–d images. *CVIU*, 112(1):81–90, 2008.

[58] Shan Zhao et al. Cellular and molecular probing of intact human organs. *Cell*, 2020.

Relationformer: A Unified Framework for *Image-to-Graph* Generation

Suprosanna Shit, Rajat Koner, Bastian Wittmann, Johannes C. Paetzold, Ivan Ezhov, Hongwei Li, Jiazhen Pan, Sahand Sharifzadeh Georgios Kaissis, Volker Tresp & Bjoern H. Menze

Conference: European Conference on Computer Vision (ECCVC), Oct 2022

Synopsis: A comprehensive representation of an image requires understanding objects and their mutual relationship, especially in *image-to-graph* generation, e.g., road network extraction, blood-vessel network extraction, or scene graph generation. Traditionally, *image-to-graph* generation is addressed with a two-stage approach consisting of object detection followed by a separate relation prediction, which prevents simultaneous object-relation interaction. This work proposes a unified one-stage transformer-based framework, namely Relationformer that jointly predicts objects and their relations. We leverage direct set-based object prediction and incorporate the interaction among the objects to learn an object-relation representation jointly. In addition to existing [`obj`]-tokens, we propose a novel learnable token, namely [`rln`]-token. Together with [`obj`]-tokens, [`rln`]-token exploits local and global semantic reasoning in an image through a series of mutual associations. In combination with the pair-wise [`obj`]-token, the [`rln`]-token contributes to a computationally efficient relation prediction. We achieve state-of-the-art performance on multiple, diverse and multi-domain datasets that demonstrate our approach's effectiveness and generalizability[1] .

[1]code is available at https://github.com/suprosanna/relationformer

8. Relationformer: A Unified Framework for *Image-to-Graph* Generation

Contributions of thesis author: Conceived the idea of Relationformer and [`rln`]-token, conceptualized a unified framework for different tasks, performed vessel-graph experiments, and took a leading role in conceptualizing the project and in organizing and writing the manuscript.

Relationformer: A Unified Framework for *Image-to-Graph* Generation

Suprosanna Shit[1,3(✉)], Rajat Koner[2], Bastian Wittmann[1], Johannes Paetzold[1], Ivan Ezhov[1], Hongwei Li[1], Jiazhen Pan[1], Sahand Sharifzadeh[2], Georgios Kaissis[1], Volker Tresp[2], and Bjoern Menze[3]

[1] Technical University of Munich, Munich, Germany
suprosanna.shit@tum.de
[2] Ludwig Maximilian University of Munich, Munich, Germany
koner@dbs.ifi.lmu.de
[3] University of Zurich, Zurich, Switzerland

Abstract. A comprehensive representation of an image requires understanding objects and their mutual relationship, especially in *image-to-graph* generation, e.g., road network extraction, blood-vessel network extraction, or scene graph generation. Traditionally, *image-to-graph* generation is addressed with a two-stage approach consisting of object detection followed by a separate relation prediction, which prevents simultaneous object-relation interaction. This work proposes a unified one-stage transformer-based framework, namely Relationformer that jointly predicts objects and their relations. We leverage direct set-based object prediction and incorporate the interaction among the objects to learn an object-relation representation jointly. In addition to existing [obj]-tokens, we propose a novel learnable token, namely [rln]-token. Together with [obj]-tokens, [rln]-token exploits local and global semantic reasoning in an image through a series of mutual associations. In combination with the pair-wise [obj]-token, the [rln]-token contributes to a computationally efficient relation prediction. We achieve state-of-the-art performance on multiple, diverse and multi-domain datasets that demonstrate our approach's effectiveness and generalizability. (Code is available at https://github.com/suprosanna/relationformer).

Keywords: Image-to-graph generation · Road network extraction · Vessel graph extraction · Scene graph generation

1 Introduction

An image contains multiple layers of abstractions, from low-level features to intermediate-level objects to high-level complex semantic relations. To gain a

S. Shit and R. Koner—Equal contribution.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19836-6_24.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13697, pp. 422–439, 2022.
https://doi.org/10.1007/978-3-031-19836-6_24

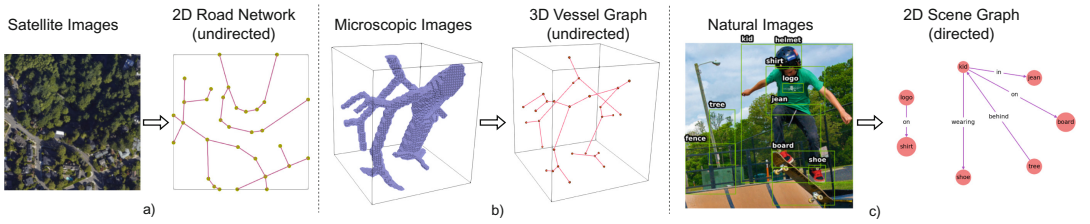

Fig. 1. Examples of relation prediction tasks. Note that the 2D road network extraction and 3D vessel graph extraction tasks have undirected relations while the scene graph generation task has directed relations.

complete visual understanding, it is essential to investigate different abstraction layers jointly. An example of such multi-abstraction problem is *image-to-graph* generation, such as road-network extraction [18], blood vessel-graph extraction [41], and scene-graph generation [55]. In all of these tasks, one needs to explore not only the objects or the *nodes*, but also their mutual dependencies or relations as *edges*.

In *spatio-structural* tasks, such as road network extraction (Fig. 1a), nodes represent road-junctions or significant turns, while edges correspond to structural connections, i.e., the road itself. The resulting spatio-structural graph construction is crucial for navigation tasks, especially with regard to autonomous vehicles. Similarly, in 3D blood vessel-graph extraction (Fig. 1b), nodes represent branching points or substantial curves, and edges correspond to structural connections, i.e., arteries, veins, and capillaries. Biological studies relying on a vascular graph representation, such as detecting collaterals [52], assessing structural robustness [21], emphasize the importance of efficient extraction thereof. In case of *spatio-semantic* graph generation, e.g. scene graph generation from natural images (Fig. 1c), the objects denote nodes and the semantic-relation denotes the edges [22]. This graphical representation of natural images is compact, interpretable, and facilitates various downstream tasks like visual question answering [19,25]. Notably, different image-to-graph tasks have been addressed separately in previous literature (see Sect. 2), and to the best of our knowledge, no unified approach has been reported so far.

Traditionally, image-to-graph generation has been studied as a complex multistage pipeline, which consist of an object detector [43], followed by a separate relation predictor [24,32]. Similarly, for spatio-structural graph generation, the usual first stage is segmentation, followed by a morphological operation on binary data. While a two-stage *object-relation* graph generation approach is modular, it is usually trained sequentially, which increases model complexity and inference time and lacks simultaneous exploration of shared object-relation representations. Additionally, mistakes in the first stage may propagate into the later stages. It should also be noted that the two-stage approach depends on multiple hand-designed features, spatial [59], or multi-modal input [8].

We argue that a single-stage image-to-graph model with joint object and relation exploration is efficient, faster, and easily extendable to multiple downstream

tasks compared to a traditional multi-stage approach. Crucially, it reduces the number of components and simplifies the training and inference pipeline (Fig. 2). Furthermore, intuitively, a simultaneous exploration of objects and relations could utilize the surrounding context and their co-occurrence. For example, Fig. 1c depicting the "kid" "on" a "board" introduces a spatial and semantic inclination that it could be an outdoor scene where the presence of a "tree" or a "helmet", the kid might wear, is highly likely. The same notion is analogous in a spatio-structural vessel graph. Detection of a "bifurcation point" and an "artery" would indicate the presence of another "artery" nearby. The mutual co-occurrence captured in joint object-relation representation overcomes individual object boundaries and leads to a more informed big picture.

Recently, there has been a surge of one-stage models in object detection thanks to the DETR approach described in [7]. These one-stage models are popular due to the simplicity and the elimination of reliance on hand-crafted designs or features. DETR exploits a encoder-decoder transformer architecture and learns object queries or [`obj`]-token for object representation.

To this end, we propose **Relationformer**, a *unified one-stage* framework for end-to-end image-to-graph generation. We leverage set-based object detection of DETR and introduce a novel learnable token named [`rln`]-token in tandem with [`obj`]-tokens. The [`rln`]-token captures the inter-dependency and co-occurrence of low-level objects and high-level spatio-semantic relations. Relationformer directly predicts objects from the learned [`obj`]-tokens and classifies their pairwise relation from combinations of [`obj-rln-obj`]-tokens. In addition to capturing pairwise object-interactions, the [`rln`]-token, in conjunction with relation information, allows all relevant [`obj`]-tokens to be aware of the global semantic structure. These enriched [`obj`]-tokens in combination with the relation token, in turn, contributes to the relation prediction. The mutually shared representation of joint tokens serves as an excellent basis for an image-to-graph generation. Moreover, our approach significantly simplifies the underlying complex image-to-graph pipeline by only using image features extracted by its backbone.

We evaluate Relationformer across numerous publicly available datasets, namely Toulouse, 20 US Cities, DeepVesselNet, and Visual Genome, comprising 2D and 3D, directed- and undirected image-to-graph generation tasks. We achieve a new state-of-the-art for one-stage methods on Visual Genome, which is better or comparable to the two-stage approaches. We achieve state-of-the-art results on road-network extraction on the Toulouse and 20 US Cities dataset. To the best of our knowledge, this is the first image-to-graph approach working directly in *3D*, which we use to extract graphs formed by blood vessels.

2 Related Work

Transformer in Vision: In recent times, transformer-based architectures have emerged as the de-facto gold standard model for various multi-domain and multi-modal tasks such as image classification [13], object detection [7] and even its application of out-of-distribution detection [23]. DETR [7] proposed an end-to-end transformer-based object detection approach with learnable object queries

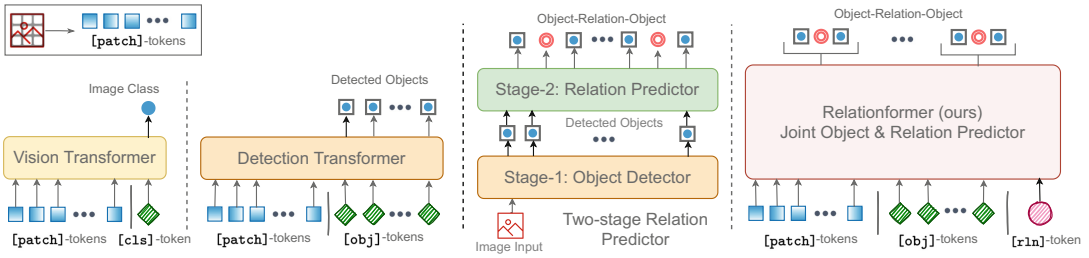

Fig. 2. This illustrates a general architectural evolution of transformers in computer vision and how Relationformer advances the concept of a task-specific learnable token one step further. The proposed Relationformer is also shown in comparison to the conventional two-stage relation predictor. The amalgamation of two separate stages not only simplifies the architectural pipeline but also co-reinforces both of the tasks.

([obj]-tokens) and direct set-based prediction. DETR eliminates burdensome object detection pipelines (e.g., anchor boxes, NMS) of traditional approaches [43] and directly predicts objects. Building on DETR, a series of object detection approaches improved DETR's slow convergence [62], adapted a pure sequence-to-sequence approach [15], and improved detector efficiency [50]. In parallel, the development of the vision transformer [13] for image classification offered a powerful alternative. Several refined idea [34,53] have advanced this breakthrough and transformer in general emerges as a cutting-edge research topic with focus on novel design principle and innovative application. Figure 2, shows a pictorial overview of transformer-based image classifier, object detector, and relation predictor including our proposed method, which we referred to as Relationformer.

Spatio-Structural Graph Generation: In a spatio-structural graph, the most important physical objects are edges, i.e., roads for a road network or arteries and veins in vessel graphs. Conventionally, spatio-structural graph extraction has only been discussed in 2D with little-to-no attention on the 3D counterpart. For 2D road network extraction, the predominant approach is to segment [4,37] followed by morphological thinning to extract the spatial graph. Only few approaches combine graph level information processing, iterative node generation [3], sequential generative modelling [9], and graph-tensor-encoding [18]. Belli et al. [5] for the first time, adopted attention mechanisms in an auto-regressive model to generate graphs conditioned on binary segmentation. Importantly, to this date, none of these 2D approaches has been shown to scale to 3D.

For 3D vessel-network extraction, segmentation of whole-brain microscopy images [39,52] has been combined with rule-based graph extraction algorithms [49]. Recently, a large-scale study [41] used the *Voreen* [38] software to extract whole-brain vascular graph from binary segmentation, which required complicated heuristics and huge computational resources. Despite recent works on 3D scene graphs [1] and temporal scene graphs [20], to this day, there exists no learning-based solution for 3D spatio-structural graph extraction.

Considering two spatio-structural image-to-graph examples of vessel-graph and road-network, one can understand the spatial relation detection task as a link prediction task. In link-prediction, graph neural networks, such as GraphSAGE

[16], SEAL [60] are trained to predict missing links among nodes using node features. These approaches predict links on a given set of nodes. Therefore, link prediction can only optimize correct graph topology. In comparison, we are interested in joint node-edge prediction, emphasizing correct topology and correct spatial location simultaneously, making the task even more challenging.

Spatio-Semantic Graph Generation: Scene graph generation (SGG) [35,55] from 2D natural images has long been studied to explore objects and their inter-dependencies in an interpretable way. Context refinement across objects [55,59], extra modality of features [35,48] or prior knowledge [46] has been used to model inter-dependencies of objects for relation prediction. RTN [24,26] was one of the first transformer approaches to explore context modeling and interactions between objects and edges for SGG. Li et al. [29] uses DETR like architecture to separately predict entity and predicate proposal followed by a graph assembly module. Later, several works [12,36] explored transformers, improving relation predictions. On the downside, such two-stage approaches increase model size, lead to high inference times, and rely on extra features such as glove vector [42] embedding or knowledge graph embedding [47], limiting their applicability. Recently, Liu et al. [33] proposed a fully convolutional one-stage SGG method. It combined a feature pyramid network [31] and a relation affinity field [40,61] for modeling a joint *object-relation* graph. However, their convolution-based architecture limits the context exploration across objects and relations. Contemporary to us [10] used transformers for the task of SGG. However, their complex pipeline for a separate subject and object further increases computational complexity. Crucially, there has been a significant performance gap between one-stage and two-stage approaches. This paper bridges this gap with simultaneous contextual exploration across objects and relations.

3 Methodology

In this section, we formally define the generalized *image-to-graph* generation problem. Each of the presented relation prediction tasks in Fig. 1 is a special instance of this generalized image-to-graph problem. Consider an image space $I \in \mathbb{R}^{D \times \#\text{ch}}$, where $D = \prod_{i=1}^{d} \dim[i]$ for a d dimensional image and #ch denotes the number of channels and $\dim[i]$ denotes the dimension of the ith spatial axis. Now, an image-to-graph generator $\mathcal{F}$ predicts $\mathcal{F}(I) = \mathcal{G}$ for a given image I, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents a graph with vertices (or objects) $\mathcal{V}$ and edges (or relations) $\mathcal{E}$. Specifically, the i^{th} vertex $v^i \in \mathcal{V}$ has a node or object location specified by a bounding box $\boldsymbol{v}^i_{\text{box}} \in \mathbb{R}^{2\times d}$ and a node or object label $v^i_{\text{cls}} \in \mathbb{Z}^C$. Similarly, each edge $e^{ij} \in \mathcal{E}$ has an edge or relation label $e^{ij}_{\text{rln}} \in \mathbb{Z}^L$, where we have C number of object classes and L types of relation classes. Note that $\mathcal{G}$ can be both directed and undirected. The algorithmic complexity of predicting graph $\mathcal{G}$ depends on its size, $|\mathcal{G}| = |\mathcal{V}| + |\mathcal{E}|$ which is of order complexity $\mathcal{O}(N^2)$ for $|\mathcal{V}| = N$. It should be noted that object detection as a special case of the generalized image-to-graph generation problem, where $\mathcal{E} = \phi$. In the following, we briefly revisit a set-based object detector before expanding on our rationale and proposed architecture.

3.1 Preliminaries of Set-Based Object Detector

Carion et al. [7] proposed DETR, which shows the potential of set-based object detection, building upon an encoder-decoder transformer architecture [54]. Given an input image I, a convolutional backbone [17] is employed to extract high level and down scaled features. Next, the spatial dimensions of extracted features are reshaped into a vector to make them sequential. Afterwards, these sequential features are coupled with a sinusoidal positional encoding [6] to mark an unique position identifier. A stacked encoder layer, consisting of a multi-head self-attention and a feed-forward network, processes the sequential features. The decoder takes N number of learnable object queries ([obj]-tokens) in the input sequence and combines them with the output of the encoder via cross-attention, where N is larger than the maximum number of objects.

DETR utilizes the direct Hungarian set-based assignment for one-to-one matching between the ground truth and the predictions from N [obj]-tokens. The bipartite matching assigns a unique predicted object from the N predictions to each ground truth object. Only matched predictions are considered valid. The rest of the predictions are labeled as $\varnothing$ or 'background'. Subsequently, it computes the box regression loss solely for valid predictions. For the classification loss, all predictions, including 'background' objects, are considered.

In our work, we adopt a modified attention mechanism, namely deformable attention from deformable-DETR (def-DETR) [62] for its faster convergence and computational efficiency. In DETR, complete global attention allows each token to attend to all other tokens and hence capture the entire context in one image. However, information about the presence of an object is highly localized to a spatial position. Following the concept of deformable convolutions [11], deformable attention enables the queries to attend to a small set of spatial features determined from learned offsets of the reference points. This improves convergence and reduces the computational complexity of the attention operation.

3.2 Object-Relation Prediction as Set-Prediction and Interaction

A joint *object-relation* graph generation requires searching from a pairwise combinatorial space of the maximum number of expected nodes. Hence, a naive joint-learning for *object-relations* requires $\mathcal{O}(N^2)$ number of tokens for N number of objects. This is computationally intractable because self-attention is quadratically-complex to the number of tokens. We overcome this combinatorially challenging task with a carefully engineered inductive bias. Here it is to exploit learned pair-wise interactions among N [obj]-tokens and combine refined pair-wise [obj]-tokens with an additional $(N+1)^{\text{th}}$ token, which we refer to as [rln]-token. One can think of the [rln]-token as a query to pair-wise object interaction.

The [rln]-token captures the additional context of pair-wise interactions among all valid predicted classes. In this process, related objects are incentivized to have a strong correlation in an embedding space of, and unrelated objects are penalized to be dissimilar. The [rln]-token attends to all N [obj]-tokens along

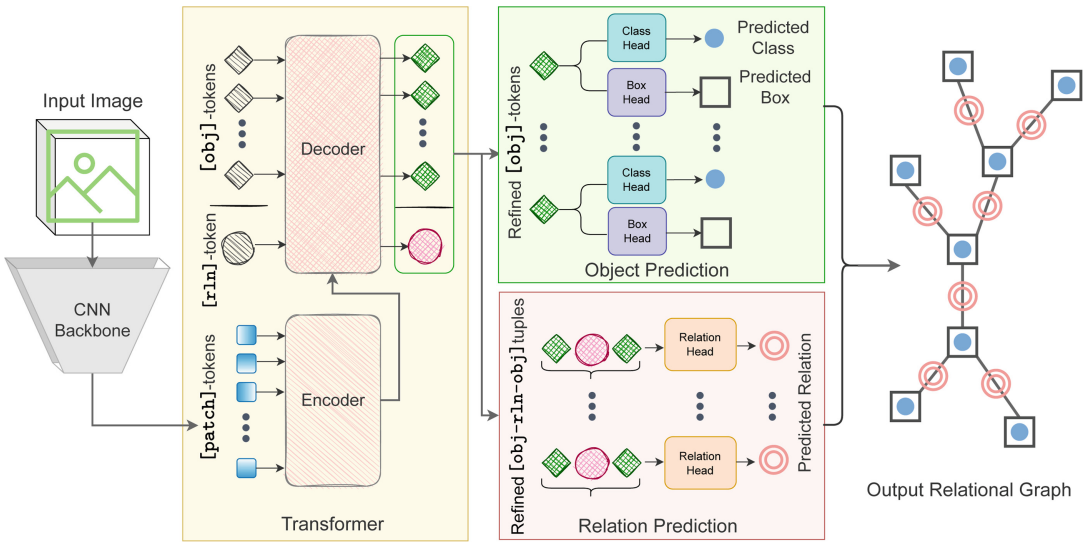

Fig. 3. Specifics of the Relationformer architecture. The image is first processed by a feature extractor, which generates [patch]-tokens for the input of the transformer encoder. Next, transformer decoder takes learnable [obj]-tokens and a [rln]-token along with output from encoder. Decoder processes them through a series of self- and cross-attention operations. The object head processes the final [obj]-tokens from the decoder to produce the bounding box and object classes. The relation head takes a tuple of the final [obj-rln-obj]-token combination and classifies their relation. Combining the output of the object and relation head yields the final graph.

with contextualized image features that enrich its local pairwise and global image reasoning. Finally, we classify a pair-wise relation by combining the pair-wise [obj]-tokens with the [rln]-token. Thus, instead of $\mathcal{O}(N^2)$ number of tokens, we only need $N + 1$ tokens in total. These consist of N [obj]-tokens and one [rln]-token. This novel formulation allows relation detection with a marginally increased cost compared to one-stage object detection.

Here, one could present a two-fold argument: 1) There is no need for an extra token as one could directly classify joint pairwise [obj]-tokens. 2) Instead of one single [rln]-token, one could use as many as all object-pairs. To answer the first question, we argue that relations can be viewed as a higher order topological entity compared to objects. Thus, to capture inter-dependencies among the relations the model requires additional expressive capacity, which can be shared among the objects. The [rln]-token reduces the burden on the [obj]-tokens by specializing exclusively on the task of relation prediction. Moreover, [obj]-tokens can also attend to the [rln]-token and exploit a global semantic reasoning. This hypothesis is confirmed in our ablation. For the second question, we argue that individual tokens for all possible object-pairs will lead to a drastic increase in the decoder complexity, which may results in computationally intractability.

3.3 Relationformer

The *Relationformer* architecture is intuitive and without any bells and whistles, see Fig. 3. We have four main components: a backbone, a transformer, an object

detection head and a relation prediction head. In the following, we describe each of the components and the set-based loss formulations specific to joint *object-relation* graph generation in detail.

Backbone: Given the input image I, a convolutional backbone [17] extracts features $\boldsymbol{f}_I \in \mathbb{R}^{D_f \times \#\,\text{emb}}$, where D_f is the spatial dimensions of the features and # emb denotes embedding dimension. Further, this feature dimension is reduced to d_{emb}, the embedding dimension of the transformer, and flattened by its spatial size. The new sequential features coupled with the sinusoidal positional encoding [6] produce the desired sequence which is processed by the encoder.

Transformer: We use a transformer encoder-decoder architecture with deformable attention [62], which considerably speeds up the training convergence of DETR by exploiting spatial sparsity of the image features.

Encoder: Our encoder remains unchanged from [62], and uses multi-scale deformable self-attention. We use a different number of layers based on each task's requirement, which is specified in detail in the supplementary material.

Decoder: We use $N + 1$ tokens for the joint *object-relation* task as inputs to the decoder, where N represents the number of [`obj`]-tokens preceded by a single [`rln`]-token. Contextualized image features from the encoder serve as the second input of our decoder. In order to have a tractable computation and to leverage spatial sparsity, we use deformable cross-attention between the joint tokens and the image features from the encoder. The self-attention in the decoder remains unchanged. The [`obj`]-tokens and [`rln`]-token go through a series of multi-hop information exchanges with other tokens and image features, which gradually builds a hierarchical object and relational semantics. Here, [`obj`]-tokens learn to attend to specific spatial positions, whereas the [`rln`]-token learns how objects interact in the context of their semantic or global reasoning.

Object Detection Head: The object detection head has two components. The first one is a stack of fully connected network or multi layer-perceptron (MLP), which regresses the location of objects, and the second one is a single layer classification module. For each refined [`obj`]-token o^i, the object detection head predicts an object class $\tilde{v}^i_{\text{cls}} = \boldsymbol{W}_{\text{cls}}(o^i)$ and an object location $\tilde{\boldsymbol{v}}^i_{\text{box}} = \text{MLP}_{\text{box}}(o^i), \tilde{\boldsymbol{v}}^i_{\text{box}} \in [0, 1]^{2\times d}$ in parallel, where d represents the image dimension, $\boldsymbol{W}_{\text{cls}}$ is the classification layer, and MLP_{box} is an MLP. We use the normalized bounding box co-ordinate for scale invariant prediction. Note that for the spatio-structural graph, we create virtual objects around each node's center by assuming an uniform bounding box with a normalized width of Δx.

Relation Prediction Head: In parallel to the object detection head, the input of the relation head, given by a pair-wise [`obj`]-token and a shared [`rln`]-token, is processed as $\tilde{e}^{ij}_{\text{rln}} = \text{MLP}_{\text{rln}}(\{o^i, r, o^j\}_{i\neq j})$. Here, r represents the refined [`rln`]-token and MLP_{rln} a three-layer fully-connected network headed by layer normalization [2]. In the case of directional relation prediction (e.g., scene graph), the *ordering* of the object token pairs $\{o^i, r, o^j\}_{i\neq j}$ determines the direction $i \rightarrow j$.

Table 1. Brief summary of the datasets used in our experiment. For more details regarding dataset preparation, please refer to supplementary material.

Dataset	Description				Data split		
	Edge type	2D/3D	Image type	Image size	Train	Val	Test
Toulouse [5]	Undirected	2D	Binary	64×64	80k	12k	19k
20 US Cities [18]	Undirected	2D	RGB	128×128	124k	13k	25k
Synthetic vessel [51]	Undirected	3D	Grayscale	$64 \times 64 \times 64$	54k	6k	20k
Visual Genome [27]	Directed	2D	RGB	800×800	57k	5k	26k

Otherwise (e.g., road network, vessel graph), the network is trained to learn object token *order* invariance as well.

3.4 Loss Function

For object detection, we utilize a combination of loss functions. We use two standard box prediction losses, namely the ℓ_1 regression loss ($\mathcal{L}_{\text{reg}}$) and the generalized intersection over union loss ($\mathcal{L}_{\text{gIoU}}$) between the predicted $\tilde{\boldsymbol{v}}_{\text{box}}$ and ground truth $\boldsymbol{v}_{\text{box}}$ box coordinates. Besides, we use the cross-entropy classification loss ($\mathcal{L}_{\text{cls}}$) between the predicted class $\tilde{v}_{\text{cls}}$ and the ground truth class v_{cls}.

Stochastic Relation Loss: In parallel to object detection, their pair-wise relations are classified with a cross-entropy loss. Particularly, we only use predicted objects that are assigned to ground truth objects by the Hungarian matcher. When two objects have a relation, we refer to their relation as a 'valid'-relation. Otherwise, the relation is categorized as 'background'. Since 'valid'-relations are highly sparse in the set of all possible permutations of objects, computing the loss for every possible pair is burdensome and will be dominated by the 'background' class, which may hurt performance. To alleviate this, we randomly sample three 'background'-relations for every 'valid'-relation. From sampled 'valid'- and 'background'-relations, we obtain a subset $\mathcal{R}$ of size M, where $\mathcal{R} \subseteq {}^{N}P_2$. To this end, $\mathcal{L}_{\text{rln}}$ represents a classification loss for the predicted relations in $\mathcal{R}$. The total loss for simultaneous *object-relation* graph generation is defined as:

$$\begin{aligned}\mathcal{L}_{\text{total}} = \sum_{i=1}^{N} \left[\mathbb{1}_{v^i_{\text{cls}} \notin \varnothing} (\lambda_{\text{reg}} \mathcal{L}_{\text{reg}}(\boldsymbol{v}^i_{\text{box}}, \tilde{\boldsymbol{v}}^i_{\text{box}}) + \lambda_{\text{gIoU}} \mathcal{L}_{\text{gIoU}}(\boldsymbol{v}^i_{\text{box}}, \tilde{\boldsymbol{v}}^i_{\text{box}}))\right] \\ + \quad \lambda_{\text{cls}} \sum_{i=1}^{N} \mathcal{L}_{\text{cls}}(v^i_{\text{cls}}, \tilde{v}^i_{\text{cls}}) + \lambda_{\text{rln}} \sum_{\{i,j\} \in \mathcal{R}} \mathcal{L}_{\text{rln}}(e^{ij}_{\text{rln}}, \tilde{e}^{ij}_{\text{rln}}) \end{aligned} \tag{1}$$

where $\lambda_{\text{reg}}, \lambda_{\text{gIoU}}, \lambda_{\text{cls}}$ and λ_{rln} are the loss functions specific weights.

4 Experiments

Datasets: We conducted experiments on four public datasets for road network generation (20 US cities [18], Toulouse [5]), 3D synthetic vessel graph generation [51], and scene-graph generation (Visual Genome [27]). The road and vessel

Table 2. Quantitative comparison of Relationformer with the different baselines for undirected graph generation datasets. Relationformer achieves a near-perfect solution for the Toulouse dataset and improves the results on the 20 US Cities dataset over baseline models. Relationformer translates a similar trend in 3D and significantly outperforms the heuristic-based approach on the synthetic vessel dataset.

Dataset	Model	Graph-level metrics				Node det.		Edge det.	
		SMD ↓	Prec. ↑	Rec. ↑	F1 ↑	mAP ↑	mAR ↑	mAP ↑	mAR ↑
Toulouse (2D)	RNN [5]	0.04857	65.41	57.52	61.21	0.50	5.01	0.21	2.56
	GraphRNN [5]	0.02450	71.69	73.21	72.44	1.34	4.15	0.34	1.01
	GGT [5]	0.01649	86.95	79.88	83.26	2.94	13.31	1.62	9.75
	Relationformer	**0.00012**	**99.76**	**98.99**	**99.37**	**94.59**	**96.76**	**83.30**	**89.87**
20 US Cities (2D)	RoadTracer [3]	N.A.	78.00	57.44	66.16	N.A.	N.A.	N.A.	N.A.
	Seg-DRM [37]	N.A.	76.54	71.25	73.80	N.A.	N.A.	N.A.	N.A.
	Seg-Orientation [4]	N.A	75.83	68.90	72.20	N.A.	N.A.	N.A.	N.A.
	Sat2Graph [18]	N.A.	80.70	72.28	76.26	N.A.	N.A.	N.A.	N.A.
	Relationformer	0.04939	**85.28**	**77.75**	**81.34**	29.25	42.84	21.78	33.19
Synthetic Vessel (3D)	U-net [45]+heuristics	0.01982	N/A	N/A	N/A	18.94	29.81	17.88	27.63
	Relationformer	**0.01107**	N/A	N/A	N/A	**78.51**	**84.34**	**78.10**	**82.15**

*N.A. indicates scores are not readily available. † N/A indicates that the metric is not applicable.

graph generation datasets are spatio-structural with a binary node and edge classification task, while the scene-graph generation dataset is spatio-sematic and has 151 node classes and 51 edge classes, including 'background' class (Table 1).

Evaluation Metrics: Given the diversity of tasks at hand, we resort to widely-used task-specific metrics. Following is a brief description, while details can be found in the supplementary material. For *Spatio-Structural Graphs*, we use four different metrics to capture spatial similarity alongside the topological similarity of the predicted graphs. 1) *Street Mover Distance (SMD)* [5] computes a Wasserstein distance between predicted and ground truth edges; 2) *TOPO Score* [18] includes precision, recall, and F-1 score for topological mismatch; 3) *Node Detection* yields mean average precision (mAP) and mean average recall (mAR) for the node; and 4) *Edge Detection* yields mAP and mAR for the edges. For *Spatio-semantic Graphs*, the scene graph detection (SGDet) metric is the most challenging and appropriate for joint object-relation detection, because it does not need apriori knowledge about object location or class label. Hence, we compute recall@K, mean-recall@K, and no-graph constraint (ng)-recall@K for $K = \{20, 50, 100\}$ on the SGDet following Zellers et al. [59]. Further, we evaluate the quality of object detection using average precision, AP@50 (IoU = 0.5) [30].

4.1 Results

Spatio-Structural Graph Generation: In spatio-structural graph generation, both correct graph topology and spatial location are equally important. Note that the objects here are represented as points in 2D/3D space. For practical reasons, we put a hypothetical box of $\Delta x = 0.2$ at the points and treat the boxes as objects.

The Toulouse dataset poses the least difficulty as we predict a graph from a binary segmentation. We notice that existing methods perform poorly. Our method improves the SMD score by three orders of magnitude. All other metrics, such as TOPO-Score (prec., rec., and F–1), indicate near-optimal topological accuracy of our method. At the same time, our performance in node and edge mAP and mAR is vastly superior to all competing methods. For the more complex 20 U.S. cities dataset, we observe a similar trend. Note that due to the lack of existing scores from competing methods (SMD, mAP, and mAR), we only compare the TOPO scores, which we outperform by a significant margin. However, when compared to the results on the Toulouse dataset, Relationformer yields lower node detection scores on the 20 U.S. cities dataset, which can be attributed to the increased dataset complexity. Furthermore, the edge detection score also deteriorates. This is due to the increased proximity of edges, i.e., parallel roads.

For 3D data, such as vessels, no learning-based comparisons exist. Hence, we compare to the current best practice [49], which relies on segmentation, skeletonization, and heuristic pruning of the dense skeleta extracted from the binary segmentation [14]. The purpose of pruning is to eliminate redundant neighboring nodes, which is error-prone due to the voxelization of the connectivity, leading to poor performances. Table 2 clearly depicts how our method outperforms the current method. Importantly, we find that our method effortlessly translates from 2D to 3D without major modifications. Moreover, our 3D model is trained end-to-end from scratch without a pre-trained backbone. To summarize, we propose the first reliable learning-based 3D spatio-structural graph generation method and show how it outperforms existing 2D approaches by a considerable margin.

Scene Graph Generation: We extensively compare our method to numerous existing methods, which can be grouped based on three concepts. One-stage methods, two-stage methods utilizing only image features, and two-stage methods utilizing extra features. Importantly, Relationformer represents a one-stage method without the need for extra features. We find that Relationformer outperforms all one stage methods in Recall and ng-Recall despite using a simpler backbone. In terms of mean-Recall, a metric addressing dataset bias, we outperform [33] and our contemporary [10] @50 and perform close to [10] @20.

In terms of object detection performance, we achieve an AP@50 of 26.3, which is close to the best performing one- and two-stage methods, even though we use a simpler backbone. Note that the object detection performance varies substantially across multiple backbones and object detectors. For example, BGNN [28] uses X-101FPN, FCSGG [33] uses HRNetW48-5S-FPN, whereas Relationformer and its contemporary RelTR [10] use a simple ResNet50 [17] backbone.

Comparing our Relationformer to two-stage models, we outperform all models that use no extra features in all metrics. Moreover, we perform almost equal to the remaining two-stage models, which use powerful backbones [28], bi-label graph resampling [28], custom loss functions [32], and extra features such as word [24] or knowledge graph embeddings [8]. Therefore, we can claim that we achieve competitive performances without custom loss functions or extra features while

using significantly fewer parameters. We also achieve much faster processing times, measured in frames per second (FPS) (see Table 3). For example, BGNN [28], which was the top performer in a number of metrics, requires three times more parameters and is an order of magnitude slower than our method.

Table 3. Quantitative results of Relationformer in comparison with state-of-the-art methods on the Visual Genome dataset. Relationformer achieves new one-stage state-of-the-art results and bridges the performance gap with two-stage models, while reducing model complexity and inference time significantly without the need for any extra features (e.g., glove vector, knowledge graph, etc.). Importantly, Relationformer outperforms two-stage models that previously reported mean-Recall@100 and ng-Recalls.

Method		Extra Feat.	Recall			mean-Recall			ng-Recall			AP	# param (M)↓	FPS ↑
			@20	@50	@100	@20	@50	@100	@20	@50	@100	@50		
Two-Stage	MOTIFS [59]	✓	21.4	27.2	30.5	4.2	5.7	6.6	–	3 0.5	35.8	20.0	240.7	6.6
	KERN [8]	✓	22.3	27.1	–	–	6.4	–	–	**30.9**	35.8	20.0	405.2	4.6
	GPS-Net [32]	✓	22.3	28.9	**33.2**	6.9	8.7	**9.8**	–	–	–	–	–	–
	BGT-Net [12]	✓	23.1	28.6	32.2	–	–	9.6	–	–	–	–	–	–
	RTN [24]	✓	22.5	29.0	33.1	–	–	–	–	–	–	–	–	–
	BGNN [28]	✓	**23.3**	**31.0**	–	**7.5**	**10.7**	–	–	–	–	**29.0**	341.9	2.3
	GB-Net [58]	✓	–	26.3	29.9	–	7.1	8.5	–	29.3	35.0	–	–	–
	IMP+ [56]	✗	14.6	20.7	24.5	2.9	3.8	4.8	–	22.0	27.4	20.0	**203.8**	**10.0**
	G-RCNN [57]	✗	–	11.4	13.7	–	–	–	–	28.5	**35.9**	23.0	–	–
One-Stage	FCSGG [33]	✗	16.1	21.3	–	2.7	3.6	–	16.7	23.5	29.2	**28.5**	87.1	8.3*
	RelTR [10]	✗	20.2	25.2	–	**5.8**	8.5	–	–	–	–	26.4	**63.7**	16.6
	Relationformer	✗	**22.2**	**28.4**	**31.3**	4.6	**9.3**	**10.7**	**22.9**	**31.2**	**36.8**	26.3	92.9	**18.2***

#param are taken from [10]. * Frame-per-second (FPS) is computed in Nvidia GTX 1080 GPU. Note that '-' indicates that the corresponding results are not available to us.

Figure 4 shows qualitative examples for all datasets used in our experiments. Qualitative and quantitative results from both spatio-structural and spatio-semantic graph generation demonstrate the efficiency of our approach and the importance of simultaneously leveraging [obj]-tokens and the [rln]-token. Relationformer achieves benchmark performances across a diverse set of image-to-graph generation tasks suggesting its wide applicability and scalability.

4.2 Ablation Studies

In our ablation study, we focus on two aspects. First, how the [rln]-token and relation-head guide the graph generation; second, the effect of the sample size in training transformers from scratch. We select the complex 3D synthetic vessel and Visual Genome datasets for the ablation. Further ablation experiments can be found in the supplementary material.

Table 4 (**Left**) evaluate the importance of the [rln]-token and different relation-head types. First, we train def-DETR only for object detection as proposed in [7,62], second, we evaluate Relationformer w/ and w/o [rln]-token and use a linear relation classification layer (models w/o the [rln]-token use only concatenated pair-wise [obj]-tokens for relation classification). Third, we replace the linear relation head with an MLP and repeat the same w/ and w/o [rln]-tokens.

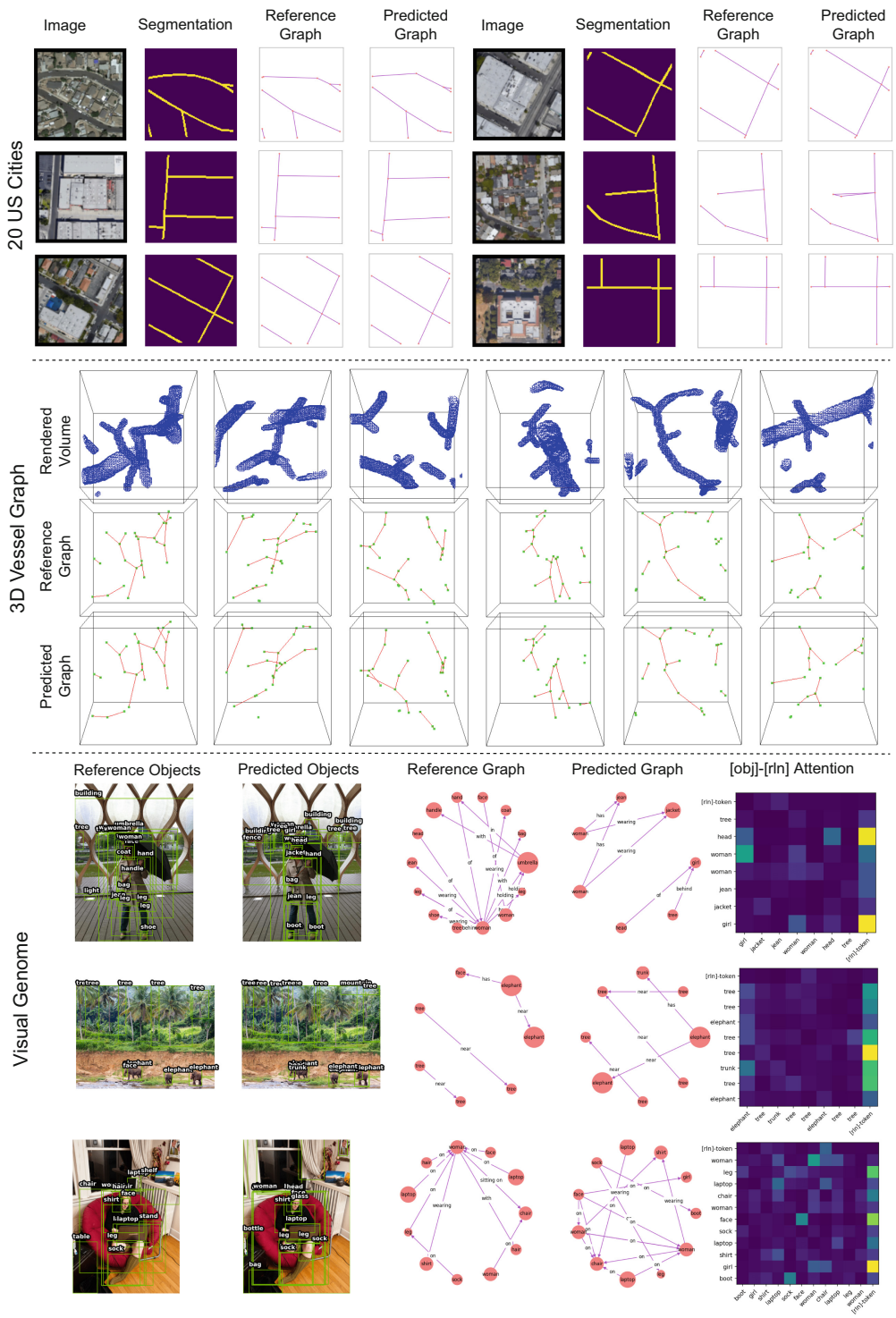

Fig. 4. Qualitative results (better viewed zoomed in) from road-network, vessel-graph, and scene-graph generation experiments. Across all datasets, we observe that Relationformer is able to produce correct results. The segmentation map is given for better interpretability of road network satellite images. For vessel-graphs, we surface-render the segmentation of corresponding greyscale voxel data. For scene graphs, we visualize the attention map between detected [`obj`]-tokens and [`rln`]-token, which shows that the [`rln`]-token actively attends to objects that contribute to relation formation.

Table 4. (**Left**) shows ablation on the [rln]-token and relation head type on Visual Genome. [rln]-token significantly improves relation prediction for both types of relation heads. Importantly, the improvement is larger for the linear classifier than for the MLP. (**Right**) shows ablation on the [rln]-token and train-data size on synthetic vessel. [rln]-token significantly improves both node and edge detection. Additionally, the scores improves with train-data size, suggesting further scope by training on more data.

Model	Visual Genome						Synthetic Vessel						
	[rln]-token	Rel. Head	AP	SGDet recall			[rln]-token	Train data	SMD	Node det.		Edge det.	
			@50	@20	@50	@100				mAP	mAR	mAP	mAR
def-DETR	N/A	N/A	26.4	N/A	N/A	N/A	N/A	100%	N/A	77.5	83.5	N/A	N/A
Relationformer	✗	Linear	24.1	16.6	22.0	25.2	✗	100%	0.0129	75.5	81.6	76.3	80.4
Relationformer	✓	Linear	25.3	20.1	25.4	28.3	✓	25%	0.0138	17.0	32.1	11.5	28.3
Relationformer	✗	MLP	26.0	19.2	26.4	29.4	✓	50%	0.0124	39.2	53.5	33.3	48.9
Relationformer	✓	MLP	26.3	22.2	28.4	31.3	✓	100%	**0.0110**	**78.5**	**84.3**	**78.1**	**82.1**

We observe that a linear relation classifier w/o [rln]-token is insufficient to model the mutual relationships among objects and diminishes the object detection performance as well. In contrast, we see that the [rln]-token significantly improves performance despite using a linear relation classifier. Using an MLP instead of a linear classifier is a better strategy whereas the Relationformer w/[rln]-token shows a clear benefit. Unlike the linear layer, we hypothesize that the MLP provides a separate and adequate embedding space to model the complex semantic relationships for [obj]-tokens and our [rln]-token.

From ablation on 3D vessel (Table 4 (**Right**)), we draw the same conclusion that [rln]-token significantly improve over Relationformer w/o [rln]-token. Further, a high correlation between performance and train-data size indicates scope for improvement by increasing the sample size while training from scratch.

Limitations and Outlook: In this work, we only use bipartite object matching, and future work will investigate graph-based matching [44]. Additionally, incorporating recent transformer-based backbones, i.e., Swin-transformer [34] could further boost the performance without compromising the simplicity.

5 Conclusion

Extraction of structural- and semantic-relational graphs from images is the key for image understanding. We propose Relationformer, a unified single-stage model for direct *image-to-graph* generation. Our method is intuitive and easy to interpret because it is devoid of any hand-designed components. We show consistent performance improvement across multiple *image-to-graph* tasks using Relationformercompared to previous methods; all while being substantially faster and using fewer parameters which reduce energy consumption. Relationformer opens up new possibilities for efficient integration of a *image-to-graph* models to downstream applications in an end-to-end fashion. We believe that our method has the potential to shed light on many previously unexplored domains and can lead to new discoveries, especially in 3D.

Acknowledgement. Suprosanna Shit is supported by TRABIT (EU Grant: 765148). Rajat Koner is funded by the German Federal Ministry of Education and Research (BMBF, Grant no. 01IS18036A). Bjoern Menze gratefully acknowledges the support of the Helmut Horten Foundation.

References

1. Armeni, I., et al.: 3D scene graph: a structure for unified semantics, 3D space, and camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5664–5673 (2019)
2. Ba, J.L., et al.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Bastani, F., et al.: RoadTracer: automatic extraction of road networks from aerial images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4720–4728 (2018)
4. Batra, A.: Improved road connectivity by joint learning of orientation and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10385–10393 (2019)
5. Belli, D., Kipf, T.: Image-conditioned graph generation for road network extraction. arXiv preprint arXiv:1910.14388 (2019)
6. Bello, I., et al.: Attention augmented convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3286–3295 (2019)
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 213–229. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
8. Chen, T., et al.: Knowledge-embedded routing network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6163–6171 (2019)
9. Chu, H., et al.: Neural turtle graphics for modeling city road layouts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4522–4530 (2019)
10. Cong, Y., et al.: RelTR: relation transformer for scene graph generation. arXiv preprint arXiv:2201.11460 (2022)
11. Dai, J., et al.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 764–773 (2017)
12. Dhingra, N., Ritter, F., Kunz, A.: BGT-Net: bidirectional GRU transformer network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2150–2159 (2021)
13. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
14. Drees, D., Scherzinger, A., Hägerling, R., Kiefer, F., Jiang, X.: Scalable robust graph and feature extraction for arbitrary vessel networks in large volumetric datasets. arXiv preprint arXiv:2102.03444 (2021)
15. Fang, Y., et al.: You only look at one sequence: rethinking transformer in vision through object detection. arXiv preprint arXiv:2106.00666 (2021)
16. Hamilton, W.L., et al.: Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 1025–1035 (2017)
17. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

18. He, S., et al.: Sat2Graph: road graph extraction through graph-tensor encoding. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12369, pp. 51–67. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58586-0_4
19. Hildebrandt, M., et al.: Scene graph reasoning for visual question answering. arXiv preprint arXiv:2007.01072 (2020)
20. Ji, J., et al.: Action genome: actions as compositions of spatio-temporal scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10236–10247 (2020)
21. Ji, X., et al.: Brain microvasculature has a common topology with local differences in geometry that match metabolic load. Neuron **109**(7), 1168–1187 (2021)
22. Johnson, J., et al.: Image retrieval using scene graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3668–3678 (2015)
23. Koner, R., Sinhamahapatra, P., Roscher, K., Günnemann, S., Tresp, V.: OOD-former: out-of-distribution detection transformer. arXiv preprint arXiv:2107.08976 (2021)
24. Koner, R., et al.: Relation transformer network. arXiv preprint arXiv:2004.06193 (2020)
25. Koner, R., Li, H., Hildebrandt, M., Das, D., Tresp, V., Günnemann, S.: Graphhopper: multi-hop scene graph reasoning for visual question answering. In: Hotho, A., et al. (eds.) ISWC 2021. LNCS, vol. 12922, pp. 111–127. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88361-4_7
26. Koner, R., et al.: Scenes and surroundings: scene graph generation using relation transformer. arXiv preprint arXiv:2107.05448 (2021)
27. Krishna, R., et al.: Visual genome: connecting language and vision using crowdsourced dense image annotations. arXiv preprint arXiv:1602.07332 (2016)
28. Li, R., et al.: Bipartite graph network with adaptive message passing for unbiased scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11109–11119 (2021)
29. Li, R., et al.: SGTR: end-to-end scene graph generation with transformer. arXiv preprint arXiv:2112.12970 (2021)
30. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
31. Lin, T.Y., et al.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
32. Lin, X., et al.: GPS-Net: graph property sensing network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3746–3753 (2020)
33. Liu, H., et al.: Fully convolutional scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11546–11556 (2021)
34. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
35. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 852–869. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_51

36. Lu, Y., et al.: Context-aware scene graph generation with Seq2Seq transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15931–15941 (2021)
37. Máttyus, G., et al.: DeepRoadMapper: extracting road topology from aerial images. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3438–3446 (2017)
38. Meyer-Spradow, J., et al.: Voreen: a rapid-prototyping environment for ray-casting-based volume visualizations. IEEE Comput. Graph. Appl. **29**(6), 6–13 (2009)
39. Miettinen, A., et al.: Micrometer-resolution reconstruction and analysis of whole mouse brain vasculature by synchrotron-based phase-contrast tomographic microscopy. BioRxiv (2021)
40. Newell, A., Deng, J.: Pixels to graphs by associative embedding. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
41. Paetzold, J.C., et al.: Whole brain vessel graphs: a dataset and benchmark for graph learning and neuroscience. In: Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2021)
42. Pennington, J., et al.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
43. Ren, S., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
44. Rolínek, M., Swoboda, P., Zietlow, D., Paulus, A., Musil, V., Martius, G.: Deep graph matching via blackbox differentiation of combinatorial solvers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12373, pp. 407–424. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58604-1_25
45. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
46. Sharifzadeh, S., et al.: Classification by attention: scene graph classification with prior knowledge. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 5025–5033 (2021)
47. Sharifzadeh, S., et al.: Improving scene graph classification by exploiting knowledge from texts. arXiv preprint arXiv:2102.04760 (2021)
48. Sharifzadeh, S., et al.: Improving visual relation detection using depth maps. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 3597–3604. IEEE (2021)
49. Shit, S., 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, pp. 16560–16569 (2021)
50. Song, H., et al.: ViDT: an efficient and effective fully transformer-based object detector. arXiv preprint arXiv:2110.03921 (2021)
51. Tetteh, G., et al.: DeepVesselNet: vessel segmentation, centerline prediction, and bifurcation detection in 3-D angiographic volumes. Front. Neurosci. **14**, 1285 (2020)
52. Todorov, M.I., et al.: Machine learning analysis of whole mouse brain vasculature. Nat. Methods **17**(4), 442–449 (2020)
53. Touvron, H., et al.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning, pp. 10347–10357. PMLR (2021)

54. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
55. Xu, D., et al.: Scene graph generation by iterative message passing. In: Computer Vision and Pattern Recognition (CVPR) (2017)
56. Xu, D., et al.: Scene graph generation by iterative message passing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5410–5419 (2017)
57. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph R-CNN for scene graph generation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11205, pp. 690–706. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_41
58. Zareian, A., Karaman, S., Chang, S.-F.: Bridging knowledge graphs to generate scene graphs. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12368, pp. 606–623. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58592-1_36
59. Zellers, R., et al.: Neural motifs: scene graph parsing with global context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5831–5840 (2018)
60. Zhang, M., Chen, Y.: Link prediction based on graph neural networks (2018)
61. Zhou, X., et al.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
62. Zhu, X., et al.: Deformable DETR: deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)

PART III

CONCLUDING REMARKS

Discussion

This dissertation dedicatedly focuses on the methodological advancement of structural and functional analysis of vascular images. In this pursuit, the core contributions are presented in Chapter 5-8 in the form of four first-authored peer-reviewed conferences and journal publications. The chapters are self-contained and have their respective discussion section. In the spirit of this cumulative thesis, I will limit the discussion to prevalent challenges, key findings, and current limitation below. Chapter 5 and 6 consider functional analysis from 4D-Flow MRI images, where physical constraints for velocity and pressure distribution are studied into the deep models. Chapter 7 and 8 deal with structural analysis of images and consider topological and graph constraints while developing the methodology.

Chapter 5 - Velocity-To-Pressure (V2P) - Net: Inferring Relative Pressures from Time-Varying 3D Fluid Flow Velocities

Pressure distribution inside critical blood vessels is an important bio-marker for neurovascular pathogens, which can be computed from measured velocity. We provided a hybrid network architecture for this computation and tried to hit the sweet spot between the efficiency of CNN and the geometry specific customization of neural implicit functions. We introduced a general framework to linearly disentangle the velocity component responsible for pressure gradient. Both the CNN and the MLP proved to be helpful in modeling different sources of pressure gradient by our proposed linear disentanglement. The CNN component of our model is generalizable across different geometry. However, the MLP has to be optimized for each geometry, which remains a main limitation of the current work. Nevertheless, the proposed strategy showed promising results and shades lights on the potential of physics informed deep models. To further improve the estimation accuracy of hemodynamic quantities, adequate resolution of the flow velocity is necessary, which is studied in the next chapter. Nevertheless, the proposed solution is based on a generic framework, which in principle could be adopted into similar problem.

Chapter 6 - SRflow: Deep Learning Based Super Resolution of 4D-flow MRI Data

Next, we looked into the super-resolution problem of the velocity field, which is needed for stable and accurate hemodynamic quantity estimation. Traditional regression loss for super-resolution considers a generic mean-squared-error loss on the velocities without dedicated emphasis on its direction. Notably, the asymmetry in the value of magnitude and directional loss makes it challenging to combine them effectively. We proposed an alternative approach to emphasize the direction of predicted velocity with respect to the reference velocity during training by computing projections onto each other. This, in turn, converts the loss surface for each velocity into an inclined one according to their direction. The experiment shows consistent improvement when the loss is used in conjunction with standard ℓ_1 loss. A drawback is the relatively low performance of standalone usage of the proposed loss, which we attribute to the decreased sensitivity of the loss at the exploitation phase of the optimization.

Chapter 7 - clDice - a Novel Topology-Preserving Loss Function for Tubular Structure Segmentation

For vessel segmentation, its connectivity is the most crucial aspect. Usual voxel-wise losses do not emphasize the connectivity mainly because lack of metric and differentiable loss to support them. We bridged this gap by focusing on the skeleton of segmentation than the segmentation itself. Checking the inclusivity of the predicted skeleton onto reference segmentation and vice versa results in a metric to efficiently capture connectivity. Theoretically, we showed that our formulation is sufficient to preserve the network topology. Further, this can be approximated in a differentiable fashion to use as a loss. Our experiments showed improved connectivity not only in the 3D vessel segmentation task but also in other similar 2D tasks. A limitation of the current realization of the differentiable skeleton is that it is an approximation of the true skeleton. Nevertheless, our solution is the most efficient compared to existing topology preserving methods till date and opens up the possibility to apply a stricter criteria for skeletonization.

Chapter 8 - Relationformer: A Unified Framework for *Image-to-Graph* Generation

Next, we look into the direct graph extraction from vascular images without the intermediate segmentation stages. This calls for a paradigm shift in the network

architecture. We bridge the voxel representation in image space with the graph representation in the vessel network space with the model set-based prediction of transformers. We borrowed concepts from the computer vision community for a single-stage predictor and proposed a unified architecture, which is applicable not only to vessel extraction but also to similar tasks in computer vision, such as road-network extraction and scene-graph extraction. Our generic architecture advances task-specific token learning in the vision transformer setup. A limitation of this approach is that it still operates on the node level to find prediction-to-reference matches during the computation of loss because of its computational intractability at the graph level. Nevertheless, this showed that it is possible to predict graphs from images in an end-to-end fashion and will encourage future research.

Outlook

With modern deep learning tools, on one side, every task can be learned end-to-end thoroughly in a data-driven way. On the other hand, it comes with a few associated drawbacks, such as heavy reliance on data, poor explainability, and domain generalization. While these are ubiquitous challenges, there are some specific bottlenecks in considering various physical constraints in deep models, as presented in this thesis. Additionally, new imaging modalities to do functional and structural in the future would also call for novel tailor made solutions.

In general, the PDE constraints reduce the requirement for additional data and make the model personalized but at the cost of partly losing generalizability. Future research direction could, in principle, take advantage of the neural implicit representation field and introduce prior in the model for increased generalizability across the computation domain. Importantly, moving from voxel representation toward point cloud [29] and mesh [30] representation is also emerging as an alternative domain to analyze functional properties of vessels such as pressure and wall shear stresses.

Advancement in the skeletonization process is also highly needed to ensure correct skeletonization during training. For that, in practice, one could use re-parameterization tricks to go from real-valued distribution to discrete values and even perform otherwise non-differentiable skeletons.

Since transformer-based architectures are heavily data-hungry, strategies to reduce that are of immense interest. One such direction would be to dissect the model and determine which part is responsible for low-level feature extraction and which part took care of relational modeling. This will allow us to transfer-learn the reusable relational modeling part and only fine-tune the modality dependant components.

Further, integration of functional aspects of the network can be integrated into the graph extraction model. For example, a differentiable flow simulator [31, 32] on a graph could be integrated to provide additional functional feedback to the network. This will lead a step forward to the grand goal of modeling inter-structural and functional dependencies of the blood vessels under normal and pathogenic conditions.

Bibliography

[1] M. I. Todorov, J. C. Paetzold, O. Schoppe, G. Tetteh, S. Shit, V. Efremov, K. Todorov-Völgyi, M. Düring, M. Dichgans, M. Piraud, et al. "Machine learning analysis of whole mouse brain vasculature." In: *Nature methods* 17.4 (2020), pp. 442–449.

[2] S. He, Y. Li, Y. Feng, S. Ho, S. Ravanbakhsh, W. Chen, and B. Póczos. "Learning to predict the cosmological structure formation." In: *Proceedings of the National Academy of Sciences* 116.28 (2019), pp. 13825–13832.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." In: *arXiv preprint arXiv:2010.11929* (2020).

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. "Language models are few-shot learners." In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[5] B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler. "Deep fluids: A generative network for parameterized fluid simulations." In: *Computer graphics forum*. Vol. 38. 2. Wiley Online Library. 2019, pp. 59–70.

[6] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, et al. "Highly accurate protein structure prediction with AlphaFold." In: *Nature* 596.7873 (2021), pp. 583–589.

[7] S. Shit, I. Ezhov, J. C. Paetzold, and B. H. Menze. "Aν-Net: Automatic Detection and Segmentation of Aneurysm." In: *Proceedings of the International workshop on Cerebral Aneurysm Detection*. Springer. 2020, pp. 51–57.

[8] M. Markl, A. Frydrychowicz, S. Kozerke, M. Hope, and O. Wieben. "4D flow MRI." In: *Journal of Magnetic Resonance Imaging* 36.5 (2012), pp. 1015–1036.

[9] H. R. Ueda, A. Ertürk, K. Chung, V. Gradinaru, A. Chédotal, P. Tomancak, and P. J. Keller. "Tissue clearing and its applications in neuroscience." In: *Nature Reviews Neuroscience* 21.2 (2020), pp. 61–79.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks." In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[11] F. Rosenblatt. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[12] Y. LeCun, Y. Bengio, et al. "Convolutional networks for images, speech, and time series." In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[13] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation." In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[14] S. Hochreiter and J. Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need." In: *Advances in neural information processing systems* 30 (2017).

[16] F. Milletari, N. Navab, and S.-A. Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation." In: *2016 fourth international conference on 3D vision (3DV)*. IEEE. 2016, pp. 565–571.

[17] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein generative adversarial networks." In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[18] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. "A simple framework for contrastive learning of visual representations." In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[19] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. "Supervised contrastive learning." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 18661–18673.

[20] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. "Neural ordinary differential equations." In: *Advances in neural information processing systems* 31 (2018).

[21] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." In: *Journal of Computational physics* 378 (2019), pp. 686–707.

[22] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin. "Accelerating eulerian fluid simulation with convolutional networks." In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3424–3433.

[23] S. Shit*, I. Ezhov*, L. Mächler, J. Lipkova, J. C. Paetzold, F. Kofler, M. Piraud, and B. H. Menze. "Semi-Implicit Neural Solver for Time-dependent Partial Differential Equations." In: *arXiv preprint arXiv:2109.01467* (2021).

[24] Z. Long, Y. Lu, X. Ma, and B. Dong. "Pde-net: Learning pdes from data." In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3208–3216.

[25] M. Raissi, A. Yazdani, and G. E. Karniadakis. "Hidden fluid mechanics: A Navier-Stokes informed deep learning framework for assimilating flow visualization data." In: *arXiv preprint arXiv:1808.04327* (2018).

[26] X. Hu, F. Li, D. Samaras, and C. Chen. "Topology-preserving deep image segmentation." In: *Advances in neural information processing systems* 32 (2019).

[27] J. C. Paetzold, J. McGinnis, S. Shit, I. Ezhov, P. Büschl, C. Prabhakar, A. Sekuboyina, M. Todorov, G. Kaissis, A. Ertürk, and B. H. Menze. "Whole Brain Vessel Graphs: A Dataset and Benchmark for Graph Learning and Neuroscience." In: *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[28] R. Koner, S. Shit, and V. Tresp. "Relation transformer network." In: *arXiv preprint arXiv:2004.06193* (2020).

[29] M. Ivantsits, L. Goubergrits, J. Brüning, A. Spuler, and A. Hennemuth. "Intracranial aneurysm rupture prediction with computational fluid dynamics point clouds." In: *International workshop on Cerebral Aneurysm Detection*. Springer. 2021, pp. 104–112.

[30] G. Li, H. Wang, M. Zhang, S. Tupin, A. Qiao, Y. Liu, M. Ohta, and H. Anzai. "Prediction of 3D Cardiovascular hemodynamics before and after coronary artery bypass surgery via deep learning." In: *Communications biology* 4.1 (2021), pp. 1–12.

[31] J. Reichold, M. Stampanoni, A. L. Keller, A. Buck, P. Jenny, and B. Weber. "Vascular graph model to simulate the cerebral blood flow in realistic vascular networks." In: *Journal of Cerebral Blood Flow & Metabolism* 29.8 (2009), pp. 1429–1443.

[32] S. Shit, C. Prabhakar, J. C. Paetzold, M. J. Menten, B. Wittmann, I. Ezhov, et al. "Graflow: Neural Blood Flow Solver for Vascular Graph." In: *Geometric Deep Learning in Medical Image Analysis (Extended abstracts)*. 2022.

PART IV

APPENDICES

Supplementary Material: SRflow: Deep Learning Based Super Resolution of 4D-flow MRI Data

Supplementary Material: Deep Learning Based Super Resolution of 4D-flow MRI Data

1 QUANTITATIVE RESULT

Methods	s	**PVNR (dB)** $\uparrow$	**RMS**$_{speed}$ (ms^{-1}) $\downarrow$	$\mathcal{E}_{dir}$ $\downarrow$	**RMS**$_{div}$ (s^{-1}) $\downarrow$
Cubic Spline	$\times 2$	32.42 ± 0.491	0.0348 ± 0.00987	0.0126 ± 0.00017	0.0019 ± 0.00051
WDSR-3D	$\times 2$	37.84 ± 0.705	0.0187 ± 0.00578	0.0097 ± 0.00005	0.0015 ± 0.00041
SRflow (ℓ_1)	$\times 2$	38.12 ± 0.696	0.0182 ± 0.00563	0.0090 ± 0.00009	$\mathbf{0.0014 \pm 0.00039}$
SRflow (mp-ℓ_1)	$\times 2$	37.71 ± 0.553	0.0168 ± 0.00508	0.0086 ± 0.00008	0.0015 ± 0.00040
SRflow (opt)	$\times 2$	$\mathbf{39.14 \pm 0.629}$	$\mathbf{0.0161 \pm 0.00487}$	$\mathbf{0.0084 \pm 0.00008}$	$\mathbf{0.0014 \pm 0.00038}$
Cubic Spline	$\times 3$	27.33 ± 0.444	0.0621 ± 0.01720	0.0317 ± 0.00030	0.0023 ± 0.00063
WDSR-3D	$\times 3$	34.55 ± 0.682	0.0269 ± 0.00826	$\mathbf{0.0102 \pm 0.00008}$	0.0018 ± 0.00054
SRflow (ℓ_1)	$\times 3$	$\mathbf{35.15 \pm 0.634}$	$\mathbf{0.0254 \pm 0.00767}$	0.0093 ± 0.00008	$\mathbf{0.0015 \pm 0.00044}$
SRflow (mp-ℓ_1)	$\times 3$	34.64 ± 0.495	0.0271 ± 0.00786	0.0119 ± 0.00002	0.0019 ± 0.00054
SRflow (opt)	$\times 3$	$\mathbf{35.20 \pm 0.520}$	$\mathbf{0.0253 \pm 0.00732}$	$\mathbf{0.0102 \pm 0.00008}$	$\mathbf{0.0015 \pm 0.00042}$
Cubic Spline	$\times 4$	24.53 ± 0.394	0.0851 ± 0.02302	0.0554 ± 0.00049	0.0027 ± 0.00077
WDSR-3D	$\times 4$	33.22 ± 0.540	0.0313 ± 0.00914	0.0102 ± 0.00012	0.0018 ± 0.00050
SRflow (ℓ_1)	$\times 4$	33.50 ± 0.676	0.0304 ± 0.00930	0.0105 ± 0.00014	0.0020 ± 0.00057
SRflow (mp-ℓ_1)	$\times 4$	33.18 ± 0.520	0.0315 ± 0.00910	$\mathbf{0.0095 \pm 0.00012}$	0.0018 ± 0.00049
SRflow (opt)	$\times 4$	$\mathbf{33.87 \pm 0.642}$	$\mathbf{0.0293 \pm 0.00888}$	$\mathbf{0.0097 \pm 0.00015}$	$\mathbf{0.0017 \pm 0.00048}$

Table S1: Experiment-1 Part A: Synthetic Cerebrovascular Results

Methods	s	**PVNR (dB)** $\uparrow$	**RMS**$_{speed}$ (ms^{-1}) $\downarrow$	$\mathcal{E}_{dir}$ $\downarrow$	**RMS**$_{div}$ (s^{-1}) $\downarrow$
Cubic Spline	$\times 2$	28.37 ± 2.046	0.0274 ± 0.01348	0.0228 ± 0.01275	0.0096 ± 0.00439
WDSR-3D	$\times 2$	29.33 ± 2.227	0.0248 ± 0.01260	0.0220 ± 0.00579	$\mathbf{0.0071 \pm 0.00347}$
SRflow (ℓ_1)	$\times 2$	29.45 ± 2.202	0.0245 ± 0.01246	0.0209 ± 0.00625	$\mathbf{0.0068 \pm 0.00329}$
SRflow (mp-ℓ_1)	$\times 2$	30.01 ± 2.215	0.0226 ± 0.01148	0.0182 ± 0.00490	0.0072 ± 0.00341
SRflow (opt)	$\times 2$	$\mathbf{30.56 \pm 2.393}$	$\mathbf{0.0220 \pm 0.01149}$	$\mathbf{0.0146 \pm 0.00403}$	0.0072 ± 0.00346
Cubic Spline	$\times 3$	23.81 ± 1.831	0.0447 ± 0.02161	0.0684 ± 0.03950	0.0092 ± 0.00422
WDSR-3D	$\times 3$	26.21 ± 1.809	0.0334 ± 0.01628	0.0525 ± 0.02257	0.0071 ± 0.00353
SRflow (ℓ_1)	$\times 3$	26.99 ± 1.923	0.0312 ± 0.01544	0.0425 ± 0.01636	$\mathbf{0.0064 \pm 0.00330}$
SRflow (mp-ℓ_1)	$\times 3$	26.80 ± 2.020	0.0314 ± 0.01555	0.0425 ± 0.01318	0.0070 ± 0.00352
SRflow (opt)	$\times 3$	$\mathbf{27.36 \pm 2.014}$	$\mathbf{0.0300 \pm 0.01489}$	$\mathbf{0.0367 \pm 0.01250}$	0.0067 ± 0.00350
Cubic Spline	$\times 4$	21.31 ± 1.738	0.0583 ± 0.02795	0.1214 ± 0.06776	0.0091 ± 0.00437
WDSR-3D	$\times 4$	25.15 ± 1.637	0.0368 ± 0.01760	0.0738 ± 0.03595	0.0068 ± 0.00341
SRflow (ℓ_1)	$\times 4$	$\mathbf{25.55 \pm 1.736}$	0.0359 ± 0.01733	$\mathbf{0.0616 \pm 0.02935}$	$\mathbf{0.0063 \pm 0.00331}$
SRflow (mp-ℓ_1)	$\times 4$	25.08 ± 1.835	0.0370 ± 0.01802	0.0677 ± 0.02744	0.0068 ± 0.00352
SRflow (opt)	$\times 4$	$\mathbf{25.61 \pm 1.848}$	$\mathbf{0.0354 \pm 0.01740}$	$\mathbf{0.0611 \pm 0.02672}$	0.0066 ± 0.00352

Table S2: Experiment-1 Part B: In Vivo Cerebrovascular 4D-flow MRI Results

Performance comparison of our proposed method with the baseline model and cubic-spline-based interpolation. We compare three different loss functions in our study for the proposed network to investigate contributions each of its contributions to the vector-field super-resolution. Higher ($\uparrow$) PVNR and lower ($\downarrow$) RMS$_{speed}$, $\mathcal{E}_{dir}$ and RMS$_{div}$ indicates better performance. We pairwise report Wilcoxon signed rank

between the best performing methods (shown in bold) and the other methods for all the metrics. Methods that do not differ significantly from the best performing one (p-value> 0.001), are also reported in bold.

Methods	s	**PVNR (dB)** $\uparrow$	**RMS**$_{speed}$ (ms^{-1}) $\downarrow$	$\mathcal{E}_{dir}$ $\downarrow$	**RMS**$_{div}$ (s^{-1}) $\downarrow$
Cubic Spline	$\times 2$	23.53 ± 3.009	0.0936 ± 0.03924	0.2316 ± 0.15496	0.0131 ± 0.00872
WDSR-3D	$\times 2$	24.80 ± 2.477	0.0805 ± 0.02708	$\mathbf{0.1902 \pm 0.13379}$	$\mathbf{0.0113 \pm 0.00741}$
SRflow (ℓ_1)	$\times 2$	$\mathbf{24.82 \pm 2.481}$	0.0805 ± 0.02696	$\mathbf{0.1898 \pm 0.13372}$	$\mathbf{0.0113 \pm 0.00745}$
SRflow (mp-ℓ_1)	$\times 2$	24.81 ± 2.666	$\mathbf{0.0762 \pm 0.02561}$	0.1929 ± 0.13352	0.0136 ± 0.00872
SRflow (opt)	$\times 2$	$\mathbf{24.86 \pm 2.532}$	$\mathbf{0.0760 \pm 0.02542}$	$\mathbf{0.1892 \pm 0.13317}$	0.0130 ± 0.00835
Cubic Spline	$\times 3$	21.60 ± 3.642	0.1252 ± 0.06540	0.3096 ± 0.18966	0.0108 ± 0.00825
WDSR-3D	$\times 3$	23.17 ± 2.774	0.1016 ± 0.03806	0.2495 ± 0.16765	$\mathbf{0.0090 \pm 0.00663}$
SRflow (ℓ_1)	$\times 3$	23.16 ± 2.784	0.1021 ± 0.03857	0.2485 ± 0.16736	$\mathbf{0.0090 \pm 0.00659}$
SRflow (mp-ℓ_1)	$\times 3$	23.15 ± 2.865	$\mathbf{0.0949 \pm 0.03669}$	0.2499 ± 0.16787	0.0109 ± 0.00788
SRflow (opt)	$\times 3$	$\mathbf{23.26 \pm 2.735}$	$\mathbf{0.0983 \pm 0.03621}$	$\mathbf{0.2482 \pm 0.16734}$	$\mathbf{0.0094 \pm 0.00694}$
Cubic Spline	$\times 4$	20.55 ± 4.061	0.1476 ± 0.08548	0.3609 ± 0.20501	0.0100 ± 0.00822
WDSR-3D	$\times 4$	22.27 ± 3.031	0.1156 ± 0.04693	0.2865 ± 0.18165	$\mathbf{0.0082 \pm 0.00633}$
SRflow (ℓ_1)	$\times 4$	22.25 ± 3.065	0.1168 ± 0.04817	$\mathbf{0.2845 \pm 0.18102}$	$\mathbf{0.0082 \pm 0.00626}$
SRflow (mp-ℓ_1)	$\times 4$	$\mathbf{22.29 \pm 3.076}$	$\mathbf{0.1061 \pm 0.04399}$	0.2869 ± 0.18090	0.0102 ± 0.00782
SRflow (opt)	$\times 4$	$\mathbf{22.38 \pm 2.963}$	0.1115 ± 0.04442	$\mathbf{0.2843 \pm 0.18081}$	$\mathbf{0.0086 \pm 0.00663}$

Table S3: Experiment-2 Part A: In Vivo Cardiovascular 4D-flow MRI Results

Methods	s	**PVNR (dB)** $\uparrow$	**RMS**$_{speed}$ (ms^{-1}) $\downarrow$	$\mathcal{E}_{dir}$ $\downarrow$	**RMS**$_{div}$ (s^{-1}) $\downarrow$
Cubic Spline	$\times 2$	28.37 ± 2.046	0.0274 ± 0.01348	0.0228 ± 0.01275	0.0096 ± 0.00439
WDSR-3D	$\times 2$	30.93 ± 2.155	0.0191 ± 0.00948	0.0096 ± 0.00184	0.0107 ± 0.00460
SRflow (ℓ_1)	$\times 2$	32.20 ± 2.373	0.0182 ± 0.00912	0.0062 ± 0.00147	0.0102 ± 0.00456
SRflow (mp-ℓ_1)	$\times 2$	33.38 ± 2.678	0.0166 ± 0.00885	0.0057 ± 0.00149	$\mathbf{0.0083 \pm 0.00388}$
SRflow (opt)	$\times 2$	$\mathbf{33.52 \pm 2.703}$	$\mathbf{0.0164 \pm 0.00878}$	$\mathbf{0.0053 \pm 0.00160}$	$\mathbf{0.0083 \pm 0.00394}$
Cubic Spline	$\times 3$	23.81 ± 1.831	0.0447 ± 0.02161	0.0684 ± 0.03950	0.0092 ± 0.00422
WDSR-3D	$\times 3$	25.59 ± 1.628	0.0373 ± 0.01592	0.0190 ± 0.00450	0.0127 ± 0.00520
SRflow (ℓ_1)	$\times 3$	27.03 ± 1.855	0.0289 ± 0.01374	0.0244 ± 0.00684	0.0089 ± 0.00378
SRflow (mp-ℓ_1)	$\times 3$	30.23 ± 2.373	0.0231 ± 0.01187	0.0139 ± 0.00393	$\mathbf{0.0070 \pm 0.00336}$
SRflow (opt)	$\times 3$	$\mathbf{30.46 \pm 2.473}$	$\mathbf{0.0228 \pm 0.01188}$	$\mathbf{0.0120 \pm 0.00345}$	$\mathbf{0.0070 \pm 0.00333}$
Cubic Spline	$\times 4$	21.31 ± 1.738	0.0583 ± 0.02795	0.1214 ± 0.06776	0.0091 ± 0.00437
WDSR-3D	$\times 4$	27.82 ± 2.192	0.0296 ± 0.01497	0.0281 ± 0.00772	$\mathbf{0.0062 \pm 0.00306}$
SRflow (ℓ_1)	$\times 4$	28.22 ± 2.271	0.0288 ± 0.01463	$\mathbf{0.0236 \pm 0.00669}$	$\mathbf{0.0063 \pm 0.00310}$
SRflow (mp-ℓ_1)	$\times 4$	27.31 ± 1.501	$\mathbf{0.0277 \pm 0.01978}$	0.0346 ± 0.00749	0.0111 ± 0.00468
SRflow (opt)	$\times 4$	$\mathbf{28.30 \pm 2.321}$	$\mathbf{0.0279 \pm 0.01456}$	$\mathbf{0.0242 \pm 0.00723}$	$\mathbf{0.0067 \pm 0.00325}$

Table S4: Experiment-2 Part B: In Vivo Cerebrovascular 4D-flow MRI Results

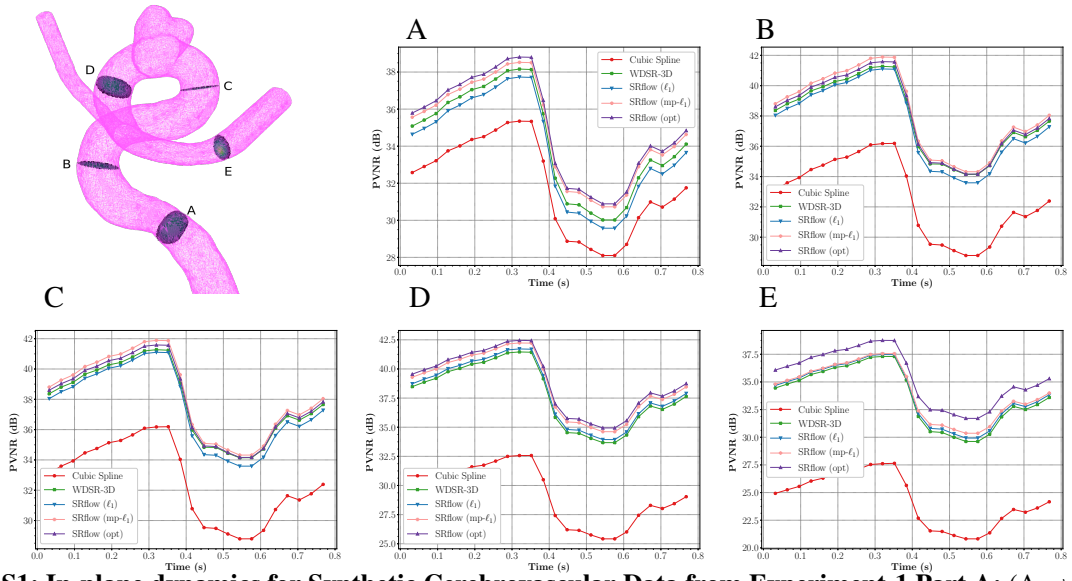

Figure S1: In-plane dynamics for Synthetic Cerebrovascular Data from Experiment-1 Part A: (A $\rightarrow$ E) shows the one cardiac cycle dynamics for PVNR for corresponding slices (A $\rightarrow$ E) of the aneurysm geometry, respectively, for the upscaling factor of $2\times$. All learning-based solutions outperform cubic-spline based super-resolution. SRflow (opt) and SRflow ($mp - \ell_1$) produces the best score in all 5 cases.

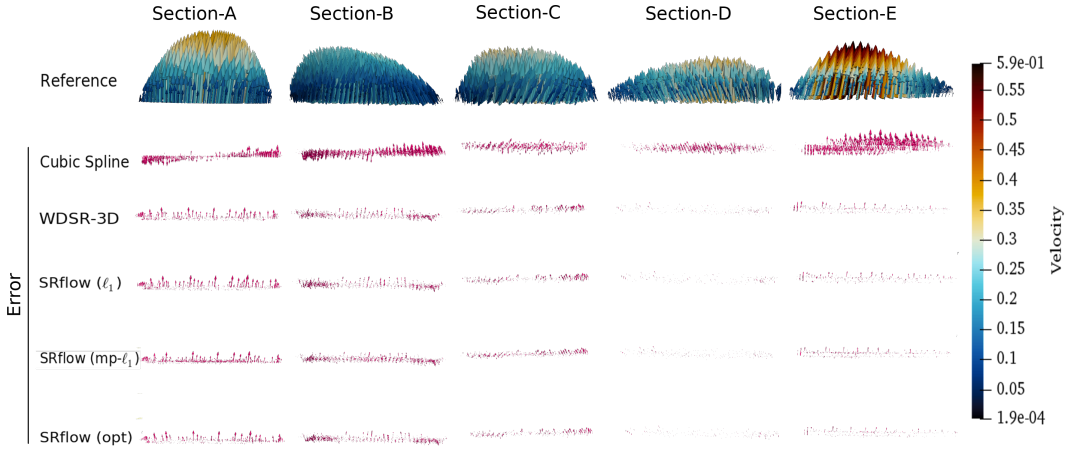

Figure S2: Flow Profile for Synthetic Cerebrovascular Data from Experiment-1 Part A: The first row shows the velocity profile of the reference data at the peak systolic time for five different cross-sections, as shown in Fig S1. The subsequent rows show the error in the velocity profile for different predictions. We observe that the cubic spline has a significant amount of error, and SRflow (opt) creates the least amount of error for all five cross-sections.

■

Supplementary Material: clDice - a Novel Topology-Preserving Loss Function for Tubular Structure Segmentation

Supplementary Material for clDice - a Novel Topology-Preserving Loss Function for Tubular Structure Segmentation

Suprosanna Shit [*1] Johannes C. Paetzold [*1] Ivan Ezhov[1] Anjany Sekuboyina[1]
Alexander Unger[1] Andrey Zhylka[2] Josien P. W. Pluim[2] Ulrich Bauer[1] Bjoern H. Menze[1]
[1]Technical University of Munich [2] Eindhoven University of Technology

1. Theory - *clDice* in Digital Topology

In addition to our Theorem 1 in the main paper, we are providing intuitive interpretations of *clDice* from the digital topology perspective. Betti numbers describe and quantify topological differences in algebraic topology. The first three Betti numbers (β_0, β_1, and β_2) comprehensively capture the manifolds appearing in 2D and 3D topological space. Specifically,

- β_0 represents the number of *connected-components*,
- β_1 represents the number of *circular holes*, and
- β_2 represents the number of *cavities* (Only in 3D)

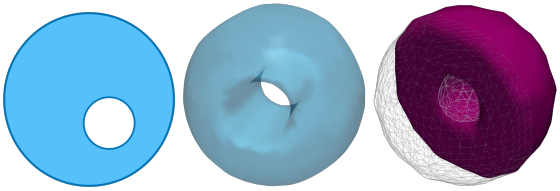

Figure 1. Examples of the topology properties. Left, a hole in 2D, in the middle a hole in 3D and right a cavity inside a sphere in 3D.

Using the concepts of Betti numbers and digital topology by Kong et al. [3, 6], we formulate the effect of topological changes between a true binary mask (V_L) and a predicted binary mask (V_P) in Fig. 2. We will use the following definition of **ghosts** and **misses**, see Figure 2.

1. **Ghosts in skeleton:** We define ghosts in the predicted skeleton (S_P) when $S_P \not\subset V_L$. This means the predicted skeleton is not completely included in the true mask. In other words, there exist false-positives in the prediction, which survive after skeletonization.

2. **Misses in skeleton:** We define misses in the predicted skeleton (S_P) when $S_L \not\subset V_P$. This means the true skeleton is not completely included in the predicted mask. In other words, there are false-negatives in the prediction, which survive after skeletonization.

The false positives and false negatives are denoted by $V_P \setminus V_L$ and $V_L \setminus V_P$, respectively, where $\setminus$ denotes a set difference operation. The loss function aims to minimize both errors. We call an error correction to happen when the value of a previously false-negative or false-positive voxel flips to a correct value. Commonly used voxel-wise loss functions, such as Dice-loss, treat every false-positive and false-negative equally, irrespective of the improvement in regards to topological differences upon their individual error correction. Thus, they cannot guarantee homotopy equivalence until and unless every single voxel is correctly classified. In stark contrast, we show in the following proposition that *clDice* guarantees homotopy equivalence under *a minimum error correction.*

Proposition 1. *For any topological differences between V_P and V_L, achieving optimal clDice to guarantee homotopy equivalence requires a minimum error correction of V_P.*

Proof. From Fig 2, any topological differences between V_P and V_L will result in ghosts or misses in the foreground or background skeleton. Therefore, removing ghosts and misses are sufficient conditions to remove topological differences. Without the loss of generalizability, we consider the case of ghosts and misses separately:

For a **ghost** $g \subset S_P, \exists$ a set of predicted voxels $E1 \subset \{V_P \setminus V_L\}$ such that $V_P \setminus E1$ does not create any misses and removes g. Without the loss of generalizability, let's assume that there is only one ghost g. Now, to remove g, under a minimum error correction of V_P, we have to minimize $|E1|$. Let's say an optimum solution $E1_{min}$ exists. By construction, this implies that $V_P \setminus E1_{min}$ removes g.

For a **miss** $m \subset V_P^{\complement}, \exists$ a set of predicted voxels $E2 \subset \{V_L \setminus V_P\}$ such that $V_P \cup E2$ does not create any ghosts and removes m. Without the loss of generalizability, let's assume that there is only one miss m. Now, to remove m, under a minimum error correction of V_P, we have to minimize $|E2|$. Let's say an optimum solution $E2_{min}$ exists. By construction, this implies that $V_P \cup E2_{min}$ removes m.

Thus, in the absence of any ghosts and misses, from Lemma 1.1, *clDice*=1 for both foreground and background. Finally, Therefore, Theorem 1 (from the main paper) guarantees homotopy equivalence. □

Lemma 1.1. *In the absence of any ghosts and misses clDice=1.*

[*]The authors contributed equally to the work

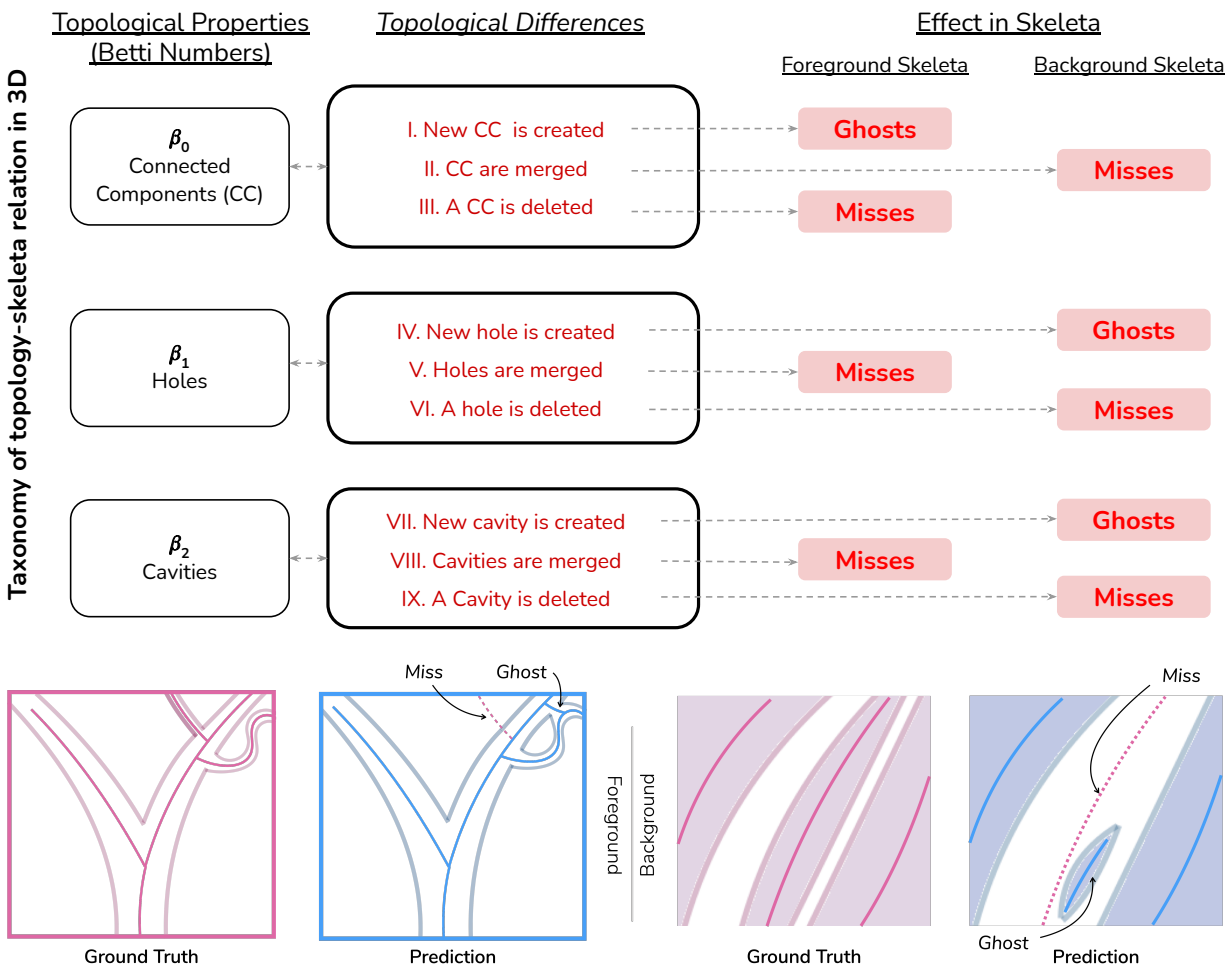

Figure 2. Upper part, left, taxonomy of the iff conditions to preserve topology in 3D using the concept of Betti numbers [3, 4]; interpreted as the necessary violation of skeleton properties for any possible topological change in the terminology of ghosts and misses (upper part right) . Lower part, intuitive depictions of ghosts and misses in the prediction; for the skeleton of the foreground (left) and the skeleton of the background (right).

Proof. The absence of any ghosts $S_P \in V_L$ implies $Tprec = 1$; and the absence of any misses $S_L \in V_P$ implies $Tsens = 1$. Hence, *clDice*=1. □

1.1. Interpretation of the Adaption to Highly Unbalanced Data According to Digital Topology:

Considering the adaptions we described in the main text, the following provides analysis on how these assumptions and adaptions are funded in the concept of ghosts and misses, described in the previous proofs. Importantly, the described adaptions are not detrimental to the performance of *clDice* for our datasets. We attribute this to the non-applicability of the necessary conditions specific to the background (i.e. II, IV, VI, VII, and IX in Figure 1), as explained below:

- II. → In tubular structures, all foreground objects are eccentric (or anisotropic). Therefore isotropic skeletonization will highly likely produce a ghost in the foreground.
- IV. → Creating a hole outside the labeled mask means adding a ghost in the foreground. Creating a hole inside the labeled mask is extremely unlikely because no such holes exist in our training data.
- VI. → The deletion of a hole without creating a miss is extremely unlikely because of the sparsity of the data.
- VII.and IX. (only for 3D) → Creating or removing a cavity is very unlikely because no cavities exist in our training data.

2. Additional Qualitative Results

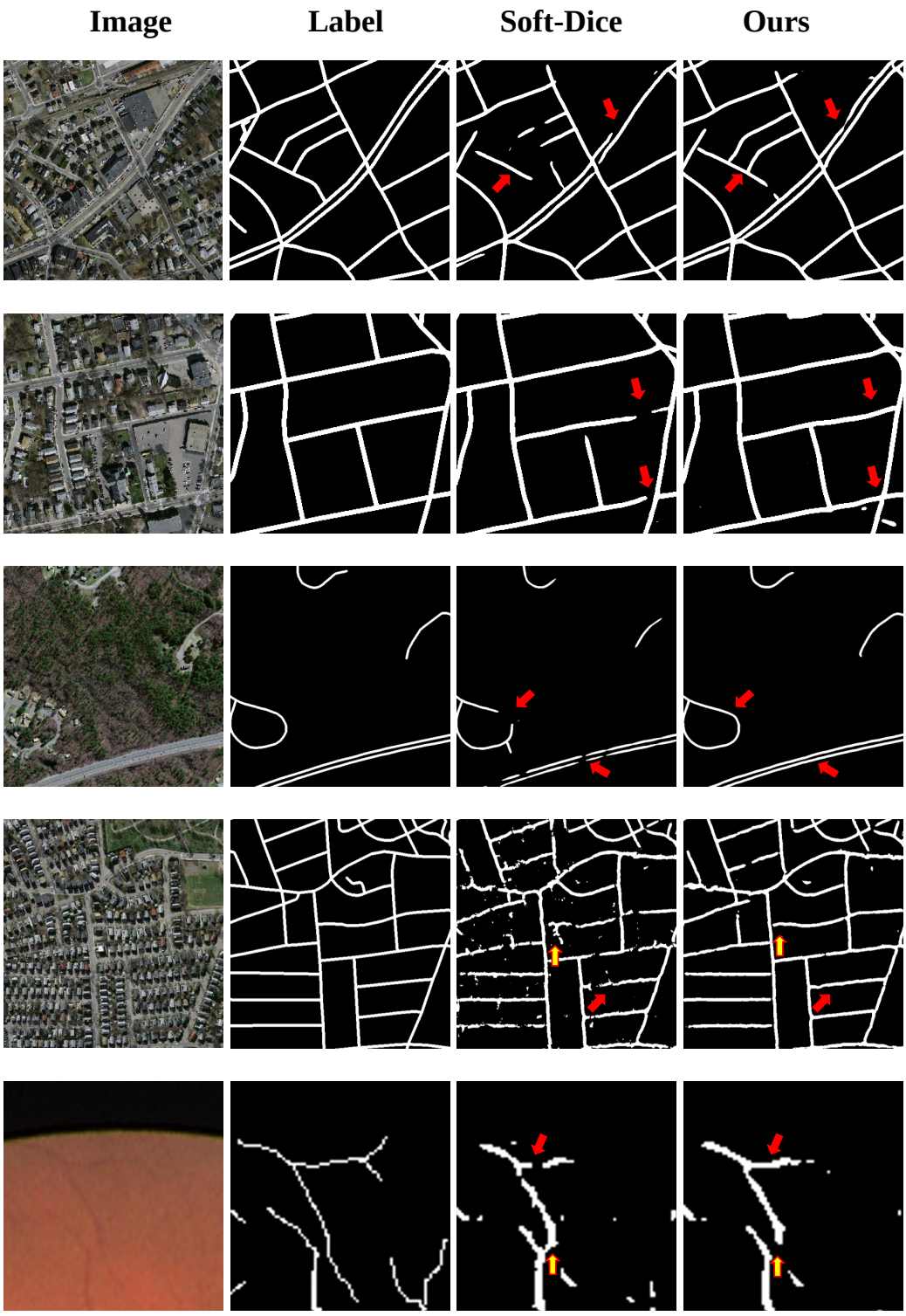

Figure 3. Qualitative results: for the Massachusetts Road dataset and for the DRIVE retina dataset (last row). From left to right, the real image, the label, the prediction using soft-dice and the predictions using the proposed $\mathcal{L}_c(\alpha = 0.5)$, respectively. The first three rows are U-Net results and the fourth row is an FCN result. This indicates that *soft-clDice* segments road connections which the soft-dice loss misses. Some, but not all, missed connections are indicated with solid red arrows, false positives are indicated with red-yellow arrows.

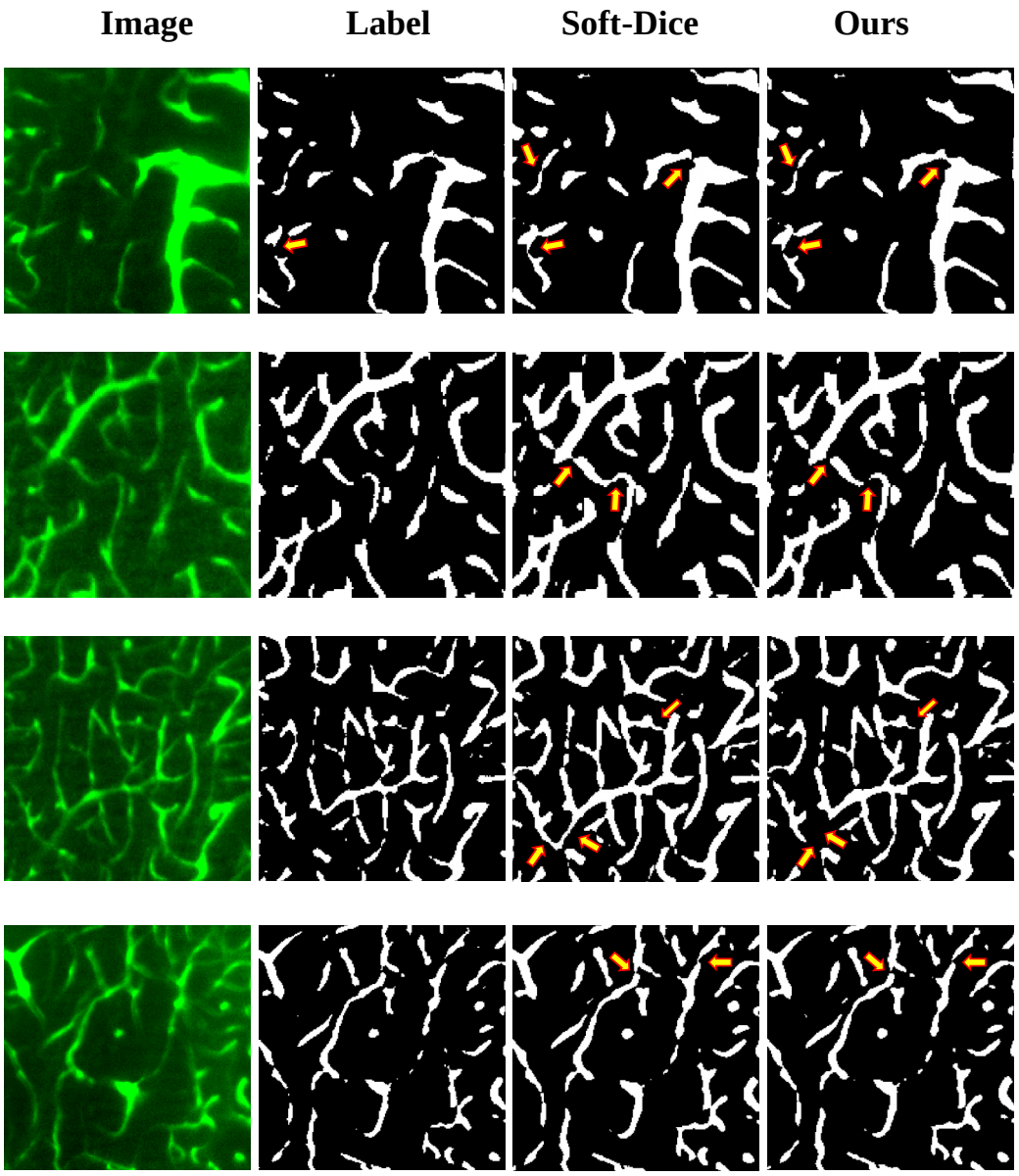

Figure 4. Qualitative results: 2D slices of the 3D vessel dataset for different sized field of views. From left to right, the real image, the label, the prediction using soft-dice and the U-Net predictions using $\mathcal{L}_c(\alpha = 0.4)$, respectively. These images show that *soft-clDice* helps to better segment the vessel connections. Importantly the networks trained using soft-dice over-segment the vessel radius and segments incorrect connections. Both of these errors are not present when we train including *soft-clDice* in the loss. Some, but not all, false positive connections are indicated with red-yellow arrows.

3. Comparison to Other Literature:

A recent pre-print proposed a region-separation approach, which aims to tackle the issue by analysing disconnected foreground elements [5]. Starting with the predicted distance map, a network learns to close ambiguous gaps by referring to a ground truth map which is dilated by a five-pixel kernel, which is used to cover the ambiguity. However, this does not generalize to scenarios with a close or highly varying proximity of the foreground elements (as is the case for e.g. capillary vessels, synaptic gaps or irregular road intersections). Any two foreground objects which are placed at a twice-of-kernel-size distance or closer to each other will potentially be connected by the trained network. This is facilitated by the loss function considering the gap as a foreground due to performing dilation in the training stage. Generalizing their approach to smaller kernels has been described as infeasible in their paper [5].

4. Datasets and Training Routine

For the DRIVE vessel segmentation dataset, we perform three-fold cross-validation with 30 images and deploy the best performing model on the test set with 10 images. For the Massachusetts Roads dataset, we choose a subset of 120 images (ignoring imaged without a network of roads) for three-fold cross-validation and test the models on the 13 official test images. For CREMI, we perform three-fold cross-validation on 324 images and test on 51 images. For the 3D synthetic dataset. we perform experiments using 15 volumes for training, 2 for validation, and 5 for testing. For the Vessap dataset, we use 11 volumes for training, 2 for validation and 4 for testing. In each of these cases, we report the performance of the model with the highest clDice score on the validation set.

5. Network Architectures

We use the following notation: $In(input\ channels)$, $Out(output\ channels)$,
$B(output\ channels)$ present input, output, and bottleneck information(for U-Net); $C(filter\ size, output\ channels)$ denote a convolutional layer followed by $ReLU$ and batch-normalization; $U(filter\ size, output\ channels)$ denote a trans-posed convolutional layer followed by $ReLU$ and batch-normalization; $\downarrow 2$ denotes maxpooling; $\oplus$ indicates concatenation of information from an encoder block. We had to choose a different FCN architecture for the Massachusetts road dataset because we realize that a larger model is needed to learn useful features for this complex task.

5.1. Drive Dataset

5.1.1 FCN :

$IN(3\text{ ch}) \rightarrow C(3,5) \rightarrow C(5,10) \rightarrow C(5,20) \rightarrow C(3,50) \rightarrow C(1,1) \rightarrow Out(1)$

5.1.2 Unet :

ConvBlock : $C_B(3, out\ size) \equiv C(3, out\ size) \rightarrow C(3, out\ size) \rightarrow \downarrow 2$

UpConvBlock: $U_B(3, out\ size) \equiv U(3, out\ size) \rightarrow \oplus \rightarrow C(3, out\ size)$

Encoder : $IN(3\text{ ch}) \rightarrow C_B(3,64) \rightarrow C_B(3,128) \rightarrow C_B(3,256) \rightarrow C_B(3,512) \rightarrow C_B(3,1024) \rightarrow B(1024)$

Decoder : $B(1024) \rightarrow U_B(3,1024) \rightarrow U_B(3,512) \rightarrow U_B(3,256) \rightarrow U_B(3,128) \rightarrow U_B(3,64) \rightarrow Out(1)$

5.2. Road Dataset

5.2.1 FCN :

$IN(3\text{ ch}) \rightarrow C(3,10) \rightarrow C(5,20) \rightarrow C(7,30) \rightarrow C(11,30) \rightarrow C(7,40) \rightarrow C(5,50) \rightarrow C(3,60) \rightarrow C(1,1) \rightarrow Out(1)$

5.2.2 Unet :

Same as Drive Dataset, except we used 2x2 up-convolutions instead of bilinear up-sampling followed by a 2D-convolution with kernel size 1.

5.3. Cremi Dataset

5.3.1 Unet :

Same as Road Dataset.

5.4. 3D Dataset

5.4.1 3D FCN :

$IN(1\text{ or }2\text{ ch}) \rightarrow C(3,5) \rightarrow C(5,10) \rightarrow C(5,20) \rightarrow C(3,50) \rightarrow C(1,1) \rightarrow Out(1)$

5.4.2 3D Unet :

ConvBlock : $C_B(3, out\ size) \equiv C(3, out\ size) \rightarrow C(3, out\ size) \rightarrow \downarrow 2$

UpConvBlock: $U_B(3, out\ size) \equiv U(3, out\ size) \rightarrow \oplus \rightarrow C(3, out\ size)$

Encoder : $IN(1\text{ or }2\text{ ch}) \rightarrow C_B(3,32) \rightarrow C_B(3,64) \rightarrow C_B(3,128) \rightarrow C_B(5,256) \rightarrow C_B(5,512) \rightarrow B(512)$

Decoder : $B(512) \rightarrow U_B(3,512) \rightarrow U_B(3,256) \rightarrow U_B(3,128) \rightarrow U_B(3,64) \rightarrow U_B(3,32) \rightarrow Out(1)$

Table 1. Total number of parameters for each of the architectures used in our experiment.

Dataset	Network	Number of parameters
Drive	FCN	15.52K
	UNet	28.94M
Road	FCN	279.67K
Cremi	UNet	31.03M
3D	FCN 2ch	58.66K
	Unet 2ch	19.21M

6. Soft Skeletonization Algorithm

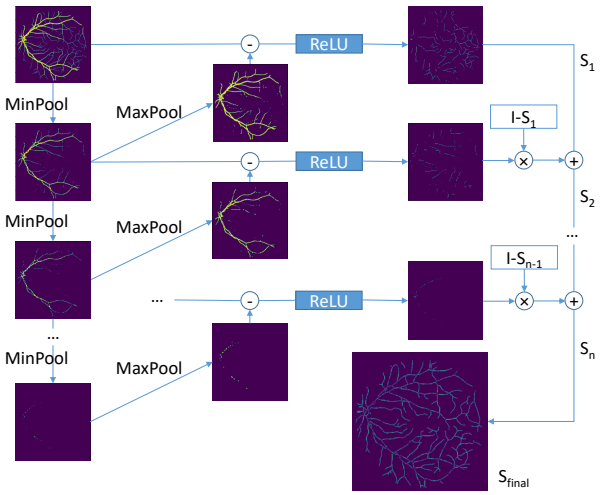

Figure 5. Scheme of our proposed differentiable skeletonization. On the top left the mask input is fed. Next, the input is reatedly eroded and dilated. The resulting erosions and dilations are compared to the image before dilation. The difference between thise images is part of the skeleton and will be added iteratively to obtain a full skeletonization. The ReLu operation eliminates pixels that were generated by the dilation but are not part of the oirginal or eroded image.

7. Code for the *clDice* similarity measure and the *soft-clDice* loss (PyTorch):

7.1. *clDice* measure

```
from skimage.morphology import skeletonize
import numpy as np
def cl_score(v, s):
    return np.sum(v*s)/np.sum(s)
def clDice(v_p, v_l):
    tprec = cl_score(v_p,skeletonize(v_l))
    tsens = cl_score(v_l,skeletonize(v_p))
    return 2*tprec*tsens/(tprec+tsens)
```

7.2. *soft-skeletonization* in 2D

```
import torch.nn.functional as F
def soft_erode(img):
    p1 = -F.max_pool2d(-img, (3,1), (1,1), (1,0))
    p2 = -F.max_pool2d(-img, (1,3), (1,1), (0,1))
    return torch.min(p1,p2)

def soft_dilate(img):
    return F.max_pool2d(img, (3,3), (1,1), (1,1))

def soft_open(img):
    return soft_dilate(soft_erode(img))

def soft_skel(img, iter):
    img1 = soft_open(img)
    skel = F.relu(img-img1)
    for j in range(iter):
        img = soft_erode(img)
        img1 = soft_open(img)
        delta = F.relu(img-img1)
        skel = skel + F.relu(delta-skel*delta)
    return skel
```

7.3. *soft-skeletonization* in 3D

```
import torch.nn.functional as F

def soft_erode(img):
    p1 = -F.max_pool3d(-img,(3,1,1),(1,1,1),(1,0,0))
    p2 = -F.max_pool3d(-img,(1,3,1),(1,1,1),(0,1,0))
    p3 = -F.max_pool3d(-img,(1,1,3),(1,1,1),(0,0,1))

    return torch.min(torch.min(p1, p2), p3)

def soft_dilate(img):
    return F.max_pool3d(img,(3,3,3),(1,1,1),(1,1,1))

def soft_open(img):
    return soft_dilate(soft_erode(img))

def soft_skel(img, iter_):
    img1 = soft_open(img)
    skel = F.relu(img-img1)
    for j in range(iter_):
        img = soft_erode(img)
        img1 = soft_open(img)
        delta = F.relu(img-img1)
        skel = skel + F.relu(delta-skel*delta)
    return skel
```

8. Evaluation Metrics

As discused in the text, we compare the performance of various experimental setups using three types of metrics: volumetric, graph-based and topology-based.

8.1. Overlap-based:

Dice coefficient, Accuracy and *clDice*, we calculate these scores on the whole 2D/3D volumes. *clDice* is calculated using a morphological skeleton (skeletonize3D from the skimage library).

8.2. Graph-based:

We extract graphs from random patches of 64×64 pixels in 2D and $48 \times 48 \times 48$ in 3D images.

For the StreetmoverDistance (SMD) [1] we uniformly sample a fixed number of points from the graph of the prediction and label, match them and calculate the Wasserstein-distance between these graphs. For the junction-based metric (Opt-J) we compute the F1 score of junction-based metrics, recently proposed by [2]. According to their paper this metric is advantageous over all previous junction-based metrics as it can account for nodes with an arbitrary number

of incident edges, making this metric more sensitive to endpoints and missed connections in predicted networks. For more information please refor to their paper.

8.3. Topology-based:

For topology-based scores we calculate the Betti Errors for the Betti Numbers β_0 and β_1. Also, we calculate the Euler characteristic, $\chi = V - E + F$, where E is the number of edges, F is the number of faces and V is the number of vertices. We report the relative Euler characteristic error (χ_{ratio}), as the ratio of the χ of the predicted mask and that of the ground truth. Note that a χ_{ratio} closer to one is preferred. All three topology-based scores are calculated on random patches of 64×64 pixels in 2D and $48 \times 48 \times 48$ in 3D images.

9. Additional Quantitative Results

Table 2. Quantitative experimental results for the 3D synthetic vessel dataset. Bold numbers indicate the best performance. We trained baseline models of binary-cross-entropy (BCE), softDice and mean-squared-error loss (MSE) and combined them with our *soft-clDice* and varied the $\alpha > 0$. For all experiments we observe that using *soft-clDice* in $\mathcal{L}_c$ results in improved scores compared to *soft-Dice*. This improvement holds for almost $\alpha > 0$. We observe that *soft-clDice* can be efficiently combined with all three frequently used loss functions.

Loss	Dice	clDice
BCE	99.81	98.24
L_c, $\alpha = 0.5$	99.76	98.25
L_c, $\alpha = 0.4$	99.77	98.29
L_c, $\alpha = 0.3$	99.76	98.20
L_c, $\alpha = 0.2$	99.78	98.29
L_c, $\alpha = 0.1$	99.82	98.39
L_c, $\alpha = 0.01$	99.83	98.46
L_c, $\alpha = 0.001$	**99.85**	**98.42**
soft-Dice	99.74	97.07
L_c, $\alpha = 0.5$	99.74	97.53
L_c, $\alpha = 0.4$	99.74	97.07
L_c, $\alpha = 0.3$	**99.80**	**98.13**
L_c, $\alpha = 0.2$	99.74	97.08
L_c, $\alpha = 0.1$	99.74	97.08
L_c, $\alpha = 0.01$	99.74	97.07
L_c, $\alpha = 0.001$	99.74	97.12
MSE	99.71	97.03
L_c, $\alpha = 0.5$	99.62	98.22
L_c, $\alpha = 0.4$	99.65	97.04
L_c, $\alpha = 0.3$	99.67	98.16
L_c, $\alpha = 0.2$	99.70	97.10
L_c, $\alpha = 0.1$	99.74	98.21
L_c, $\alpha = 0.01$	99.82	98.32
L_c, $\alpha = 0.001$	**99.84**	**98.37**

References

[1] Davide Belli and Thomas Kipf. Image-conditioned graph generation for road network extraction. *arXiv preprint arXiv:1910.14388*, 2019. 4326

[2] Leonardo Citraro, Mateusz Koziński, and Pascal Fua. Towards reliable evaluation of algorithms for road network reconstruction from aerial images. In *European Conference on Computer Vision*, pages 703–719. Springer, 2020. 4326

[3] T. Yung Kong. On topology preservation in 2-D and 3-D thinning. *International journal of pattern recognition and artificial intelligence*, 9(05):813–844, 1995. 4321, 4322

[4] T Yung Kong and Azriel Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48(3):357–393, 1989. 4322

[5] Doruk Oner, Mateusz Koziński, Leonardo Citraro, Nathan C Dadap, Alexandra G Konings, and Pascal Fua. Promoting connectivity of network-like structures by enforcing region separation. *arXiv preprint arXiv:2009.07011*, 2020. 4324

[6] Azriel Rosenfeld. Digital topology. *The American Mathematical Monthly*, 86(8):621–630, 1979. 4321

Supplementary Material: Relationformer: A Unified Framework for *Image-to-Graph* Generation

Supplementary Material
Relationformer: A Unified Framework for *Image-to-Graph* Generation

Suprosanna Shit*[1,3], Rajat Koner*[2], Bastian Wittmann[1], Johannes Paetzold[1], Ivan Ezhov[1], Hongwei Li[1], Jiazhen Pan[1], Sahand Sharifzadeh[2], Georgios Kaissis[1], Volker Tresp[2], and Bjoern Menze[3]

[1] Technical University of Munich, Munich, Germany
[2] Ludwig Maximilian University of Munich, Munich, Germany
[3] University of Zurich, Zurich, Switzerland
suprosanna.shit@tum.de, koner@dbs.ifi.lmu.de

A Transformer and Deformable-DETR

The core of a transformer [12] is the attention mechanism. Let us consider an image feature map $\boldsymbol{f}_I$, the q^{th} query with associated features $\boldsymbol{f}_q$ and k^{th} key with associated image features $\boldsymbol{f}_I^k$. One can define the multi-head attention for M number of heads and K number of key elements as

$$\text{MultiHeadAttn}\left(\boldsymbol{f}_q, \boldsymbol{f}_I\right) = \sum_{m=1}^{M} \boldsymbol{W}_m \left[\sum_{k=1}^{K} \boldsymbol{A}_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{f}_I^k\right]$$

where $\boldsymbol{W}'_m$ and $\boldsymbol{W}_m$ are learnable weights. The attention weights $\boldsymbol{A}_{mqk} \propto \exp\left\{\frac{\boldsymbol{f}_q^\top \boldsymbol{W}''^\top_m \boldsymbol{W}'''_m \boldsymbol{f}_I^k}{\sqrt{d_k}}\right\}$ are normalized as $\sum_{k=1}^{K} \boldsymbol{A}_{mqk} = 1$, where $\boldsymbol{W}''_m, \boldsymbol{W}'''_m$ are also learnable weights and d_k is the temperature parameter. To differentiate position of each element uniquely, $\boldsymbol{f}_q$ and $\boldsymbol{f}_I$ are given a distinct positional embedding.

In our work, we use the multi scale deformable attention [14]. Let us consider the reference point associated with $\boldsymbol{f}_q$ as $\boldsymbol{x}_q$. First, for the m^{th} attention head, we need to compute the k^{th} sampling offset $\Delta\boldsymbol{x}_{mqk}$ based on the query features $\boldsymbol{f}_q$. Subsequently, the sampled image features $\boldsymbol{f}_I(\boldsymbol{x}_q + \Delta\boldsymbol{x}_{mqk})$ go through a single layer $\boldsymbol{W}'_m$ followed by a multiplication with the attention weight $\boldsymbol{A}_{mqk}$, which is also obtained from the query features $\boldsymbol{f}_q$. Finally, another single layer $\boldsymbol{W}_m$ merges all the heads. Formally, the deformable attention operation (DefAttn) for M heads and K sampling points is defined as:

$$\text{DefAttn}(\boldsymbol{f}_q, \boldsymbol{x}_q, \boldsymbol{f}_I) = \sum_{m=1}^{M} \boldsymbol{W}_m \left[\sum_{k=1}^{K} \boldsymbol{A}_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{f}_I(\boldsymbol{x}_q + \Delta\boldsymbol{x}_{mqk})\right] \quad (1)$$

* equal contribution

The multi-scale deformable attention for L number of level is given as

$$\text{MSDefAttn}(\boldsymbol{f}_q, \boldsymbol{x}_q, \{\boldsymbol{f}_I^l\}_{l=1}^L) = \sum_{m=1}^{M} \boldsymbol{W}_m \left[\sum_{l=1}^{L}\sum_{k=1}^{K} \boldsymbol{A}_{mlqk} \cdot \boldsymbol{W}^{'}_m \boldsymbol{f}_I^l(\phi_l(\boldsymbol{x}_q) + \Delta \boldsymbol{x}_{mlqk})\right]$$

where ϕ_l rescales the normalized reference point coordinates appropriately in the corresponding image space.

B Dataset

Here we describe the individual datasets used in our experimentation in detail. We also elaborate on generating train-test sets for our experiments. For 20 U.S. Cities and 3D synthetic vessel we extract overlapping patches from large images. This provides us a large enough sample size to train our Relationformer from scratch. Since, a DETR like architecture is not translation invariant because of learned [`obj`]-tokens in the decoder, extracting overlapping patches drastically increases the effective sample size within a limited number of available images.

B.1 Toulouse Road Network

The Toulouse Road Network dataset [1] is based on publicly available satellite images from Open Streetmap and consists of semantic segmentation images with their corresponding graph representations. For our experiments we use the same split as in the original dataset paper with 80,357 samples in the training set, 11,679 samples in the validation set, and 18,998 samples in the test set [1].

B.2 20 U.S. Cities Dataset

For the 20 U.S. Cities dataset [3], there are 180 images with a resolution of 2048x2048. We select 144 for training, 9 for validation, and 27 for testing. From those images, we extract overlapping patches of size 128x128 to construct the final train-validation-test split. We crop the RGB image and the corresponding graph followed by a node simplification. Following Belli et al. [1], we prune the dense nodes by computing the angle between two road-segments at each node of degree 2 and only keep a node if the road curvature is less than 160 degrees. This allows eliminating redundant nodes and simplifying the graph prediction task. Fig. 1 illustrates the pruning process.

B.3 3D Synthetic Vessels

Our synthetic vessel dataset is based on publicly available synthetic images generated in Tetteh et al. [11]. In this dataset, the ground truth graph was generated by [10] and from that, corresponding voxel-level semantic segmentation data was generated. Grey valued data was obtained by adding different noise levels to the

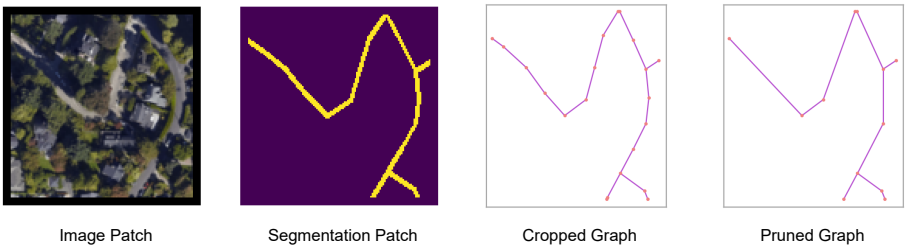

Fig. 1. Preprocessing steps for the 20 U.S. Cities dataset. The same steps are followed in the 3D Synthetic Vessel dataset curation.

segmentation map. Specifically, we train on greyscale "images" and their corresponding vessel graph representations, where each node represents a bifurcation point, and the edges represent their connecting vessels. The whole dataset contains 136 3D volumes of size 325x304x600. First, we choose 40 volumes to create a train and validation set and next pick 10 volumes for the test set. From this, we extract overlapping patches of size 64x64x64 to construct the final train-validation-test set. Similar to the 20 U.S. cities dataset, we prune nodes having degree 2 based on the angle between two edges.

B.4 Visual Genome

Visual Genome is one of the largest scene graph datasets consisting of 108,077 natural images [6]. However, the original dataset suffers from multiple annotation errors and improper bounding boxes. Lu et al. [9] proposed a refined version of Visual Genome with the most frequent occurring 150 objects classes and 50 relation categories. It also proposed its own train/val/test splits and is the most widely used data-split [13,5,7,8] for SGG. For fair comparison, we only train on the Visual Genome dataset and do **not** use any pre-training.

C Metrics Details

Metrics for Spatio-Structural Graph: We use three different kinds of metrics to capture spatial similarity alongside the topological similarity of the predicted graphs. The graph-level metrics include; 1) *Street Mover Distance (SMD):* SMD[1] compute Wasserstein distance between the uniformly sampled fixed number of points (See Fig. 2) from the predicted and ground truth edges; and 2) *TOPO Score:* TOPO Score[3] computes precision, recall, and F-1 score for topological mismatch in terms of the false-positive and false-negative topological loop. Alongside, we use 3) *Node Detection:* For this, we report mean average precision (mAP) and mean average recall (mAR) over a threshold range [0.5,0.95,0.05] for node box prediction. Similarly, we use 4) *Edge Detection:* We compute the mAP and mAR for the edge in the same way as above. The edge

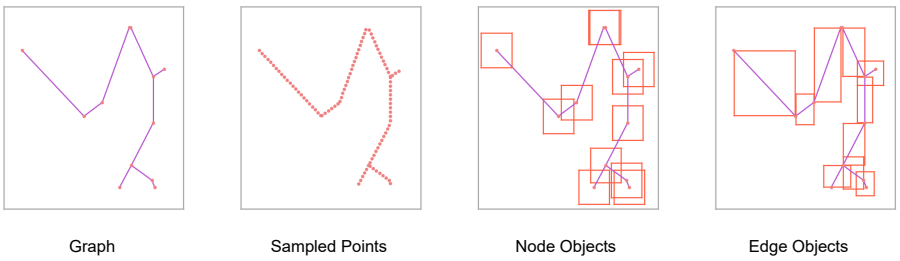

Fig. 2. Sampled points, node objects and edge objects for computing different spatio-structural graph metrics. The same notion is used for 3D graphs.

boxes are constructed from the center points of two connecting nodes (See Fig. 2). For vertical and horizontal edges we assume an hypothetical width of 0.15 to avoid objects with near zero width.

Metrics for Spatio-Semantic Graph: We evaluate Relationformer on the most challenging Scene Graph Detection(SGDet) metrics and its variants. Unlike other scene graph metrics like Predicate Classification (PredCls) or Scene Graph classification (SGCls) , SGDet does not use apriori information on class label or object spatial position and does not rely on complex RoI-align based spatial features. SGDet jointly measures the predicted boxes (with 50% overlaps) class labels of an object, and relation labels. The variants of SGDet include 1) *Recall:* Recall at the different K (20, 50 and 100) of predicted relation that reflects overall relation prediction performance, 2) *Mean-Recall:* mean-Recall computes mean of each relation class-wise recall that reflects the performance under the relational imbalance or long-tailed distribution of relation class, 3) *ng-Recall:* ng-Recall is recall w/o graph constraints on the prediction, which takes the top-k predictions instead of just the top-1. Additionally, we use 4) AP@50: Average precision at 50% threshold of IOU reflects an average object detection performance.

D Model Details

Table 1. The model parameters used in Relationformer experiments across the various datasets. Specifically, we list details on the backbone and the transformer's number of layers, feature dimension and other details.

DataSet	Backbone	Transformer				MLP Dim
		Enc. Layer	Dec. Layer	# [obj]-tokens	d_{emb}	
Toulouse	ResNet-50	4	4	20	256	512
20 US cities	ResNet-101	4	4	80	512	1024
Synth Vessel	SE-Net	4	4	80	256	1024
Visual Genome	ResNet-50	6	6	200	512	2048

Table 1, describes the backbone and important parameters of the Relationformer. We experiment with different ResNet backbones to show the flexibility of our Relationformer. In order to reduce energy consumption, we use the lighter ResNet50 for most 2D datasets. For the 3D experiment, we used Squeeze-and-Excite Net [4]. We used the number of encoder and decoder layers and the number of [obj]-tokens in the increasing order of dataset complexity. We find that four transformer layers and 20 [obj]-tokens suffice for Toulouse, while we need four transformer layers and 80 [obj]-tokens are required for 20 U.S. cities and synthetic vessel datasets. We need 6 layers of transformer and 200 [obj]-tokens for the visual genome. The ablation on the number of transformer layers and number of [obj]-tokens are shown in the next section.

E Training Details

Table 2. A list of the important set of parameters used in Relationformer for respective training. Furthermore, we list the weights for bipartite matching costs and training losses.

DataSet	Batch Size	Learning rate	Epoch	Cost Coeff.			Loss Coeff.			
				cls	reg	gIoU	λ_{reg}	λ_{gIoU}	λ_{cls}	λ_{rln}
Toulouse	64	10^{-4}	50	2	5	0	5	2	2	1
20 US cities	32	10^{-4}	100	3	5	0	5	2	3	4
3D Vessel Net	48	10^{-4}	100	2	5	0	2	3	3	4
Visual Genome	16	10^{-4}	25	3	2	3	2	2	4	6

Table. 2, summarizes some principal parameters we use in the training. We use AdamW optimizer with a step learning rate. For scene graph generation, we use the prior statistical distribution or frequency-bias [13] of relation for each subject-object pair. To minimize the data imbalance for a relation label present in the Visual Genome, we use log-softmax distribution [7] to soften the frequency bias. Finally, we add this distribution with the predicted relation distribution from the relation head. For the spatio-structural dataset, we set the cost coefficient for the GIoU in the bipartite matcher to be zero because we assume 0.2 widths of the normalized box for each node. Hence, ℓ_1 cost is sufficient to consider for the spatial distances.

F More Ablation Studies on [obj]-tokens and Transformer

We conduct two more ablation studies on Visual Genome for analyzing the influence of [obj]-tokens and optimal number of layers in transformer for the joint graph generation. Furthermore Figure. 3 gives additional insight how [rln]-token is beneficial for joint object-relation graph.

Table 3. Impact of the [obj]-tokens on joint object and relation detection.

#[obj]-tokens	AP@50	R@20	R@50	R@100
75	25.1	20.6	26.1	29.5
100	25.8	21.1	27.4	30.6
200(ours)	26.3	22.2	28.4	31.3
300	26.3	21.9	27.9	31.0

Table 4. Impact of the transformer's layers on joint object-relation detection

# layer	AP@50	R@20	R@50	R@100
4	24.6	20.5	26.5	28.8
5	25.2	21.0	27.2	29.9
6(ours)	26.3	22.2	28.4	31.3

As shown in Table 3, it can be observed that increasing [obj]-tokens does increase object and relation detection performance. However, it becomes relatively stable with increasing object quarries. DETR-like architectures rely on an optimal number of [obj]-tokens to balance positive and negative simple which also helps in object detection as observed in [2]. Thus, in a joint object and relation prediction, a gain might come from optimal number [obj]-tokens, as relation prediction is linearly co-related to object detection performance. It demonstrates that joint object and relation detection can perfectly coexist without hurting the object detection performance. Instead, it can exploit [obj]-tokens enriched with global relational reasoning for efficient relation extraction.

During the ablation with transformer layers, we observe decreasing number of transformer layers shows an initial gain in object and relation detection. However, they lead to early plateau and inferior performance as depicted in table 4. One intuitive reason is that with less parameter and insufficient contextualization Relationformer quickly learn the initial biases present in both object and relation detection and failed to learn the complex global scenario. We use the same number of layers for both encoder and decoder.

G Qualitative Results

Fig. 4 and 5 shows additional qualitative example from our experiments.

References

1. Belli, D., Kipf, T.: Image-conditioned graph generation for road network extraction. arXiv preprint arXiv:1910.14388 (2019)
2. Carion, N., et al.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
3. He, S., et al.: Sat2Graph: road graph extraction through graph-tensor encoding. In: European Conference on Computer Vision. pp. 51–67. Springer (2020)
4. Hu, J., et al.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
5. Koner, R., et al.: Relation transformer network. arXiv preprint arXiv:2004.06193 (2020)
6. Krishna, R., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. arXiv preprint arXiv:1602.07332 (2016)
7. Lin, X., et al.: Gps-net: Graph property sensing network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3746–3753 (2020)
8. Liu, H., et al.: Fully convolutional scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11546–11556 (2021)
9. Lu, C., et al.: Visual relationship detection with language priors. In: European Conference on Computer Vision (2016)
10. Schneider, M., et al.: Tissue metabolism driven arterial tree generation. Med Image Anal. **16**(7), 1397–1414 (2012)
11. Tetteh, G., et al.: Deepvesselnet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes. Frontiers in Neuroscience **14**, 1285 (2020)
12. Vaswani, A., et al.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
13. Zellers, R., et al.: Neural motifs: Scene graph parsing with global context. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5831–5840 (2018)
14. Zhu, X., et al.: Deformable DETR: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)

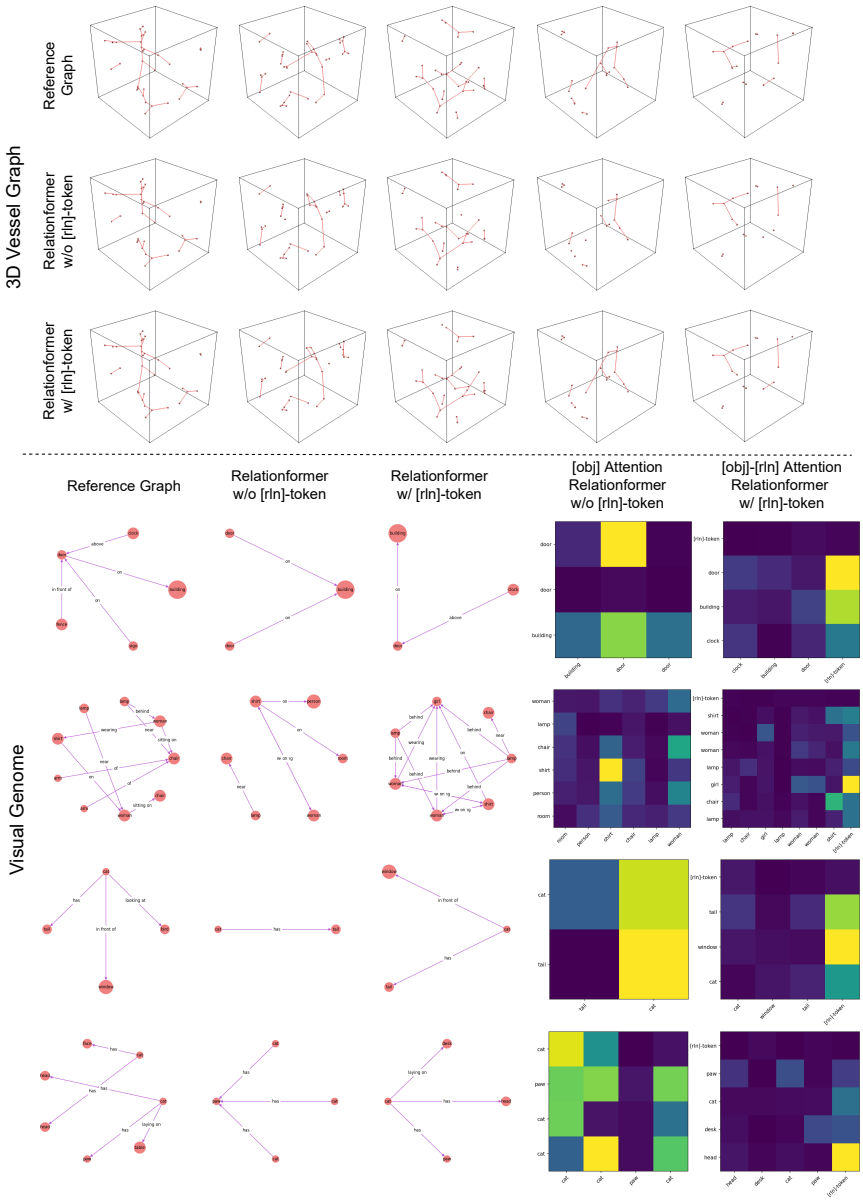

Fig. 3. Typical qualitative results (please zoom in) from our ablation on the synthetic vessel-graph and visual genome datasets. We observe that Relationformer w/o [`rln`]-token is missing vessel edges while Relationformer w/ [`rln`]-token produces correct edges. For visual genome, we can see w/o [`rln`]-token the [`obj`]-tokens have to carry extra burden for relation prediction and sometimes fail to incorporate the global relation. However, the inclusion of [`rln`]-token provides an additional path to flow relation information that benefits the joint object and relation detection.

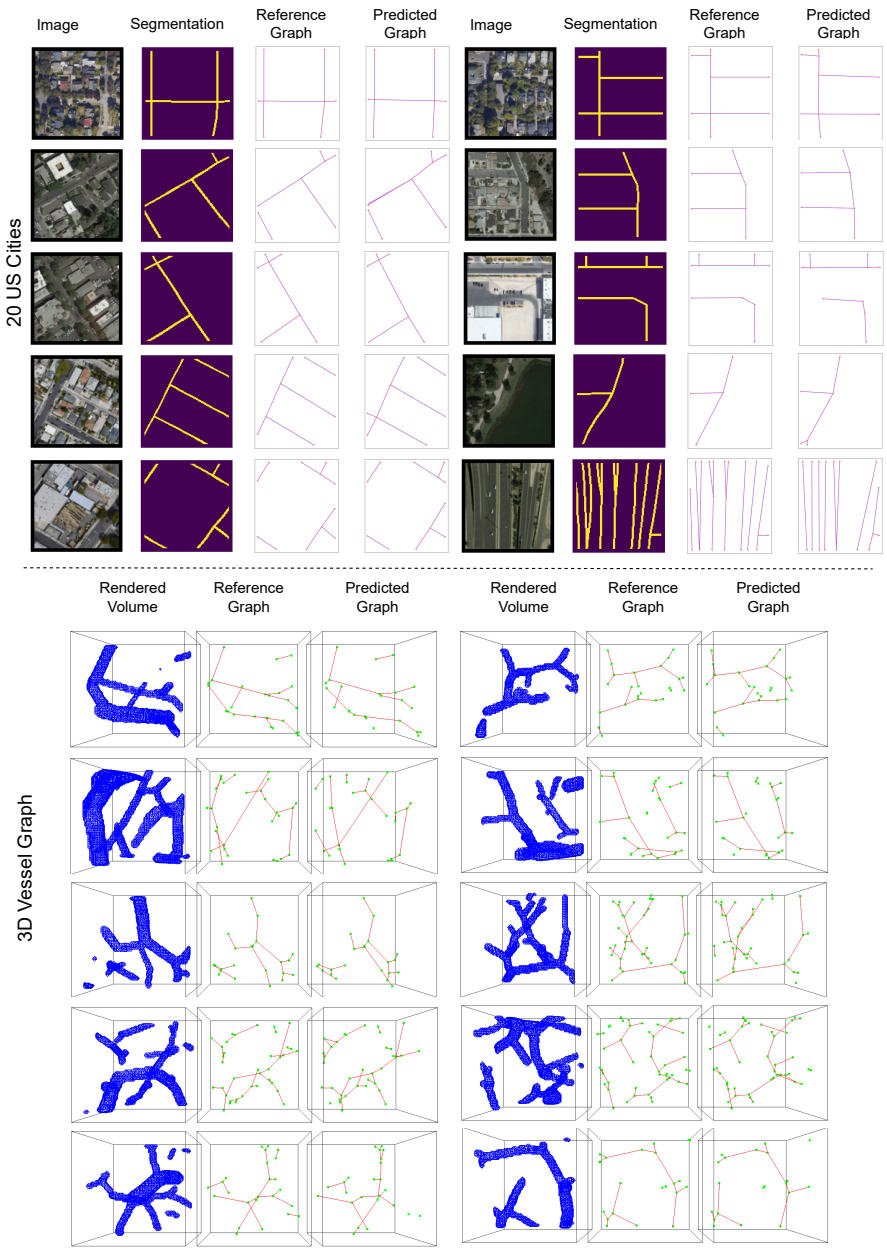

Fig. 4. Qualitative results (please zoom in) for the 20 US cities road-network and synthetic vessel-graph experiments. We observe that Relationformer is able to produce correct results. The segmentation map is given for better interpretability of road network satellite images. For vessel-graphs, we surface-render the segmentation of the corresponding greyscale voxel data.

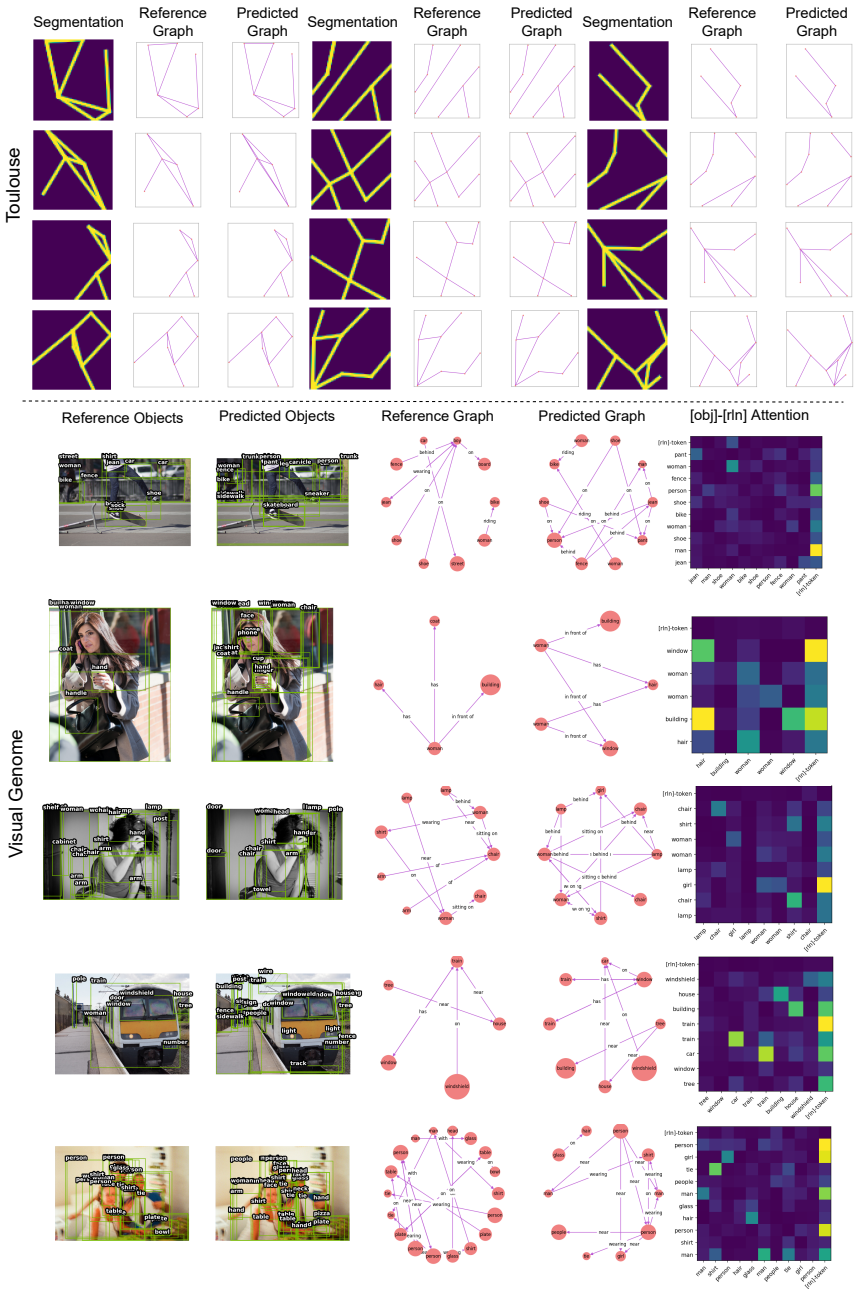

Fig. 5. Qualitative results (please zoom in) from the Toulouse road-network and scene-graph generation experiments. For both datasets, we observe that Relationformer is able to generate an accurate graph. For scene graphs, we visualize the attention map between detected [`obj`]-tokens and [`rln`]-token, which shows that the [`rln`]-token actively attends to objects that contribute to relation formation.

Aν-net: Automatic Detection and Segmentation of Aneurysm

Aν-Net: Automatic Detection and Segmentation of Aneurysm

Suprosanna Shit[1,2( )], Ivan Ezhov[1,2], Johannes C. Paetzold[1,2,3], and Bjoern Menze[1,2,4]

[1] Departments of Informatics, Technical University Munich, Munich, Germany
suprosanna.shit@tum.de
[2] TranslaTUM Center for Translational Cancer Research, Munich, Germany
[3] Institute for Tissue Engineering and Regenerative Medicine Helmholtz Zentrum München, Neuherberg, Germany
[4] Department of Quantitative Biomedicine of UZH, Zurich, Switzerland

Abstract. We propose an automatic solution for the CADA 2020 challenge to detect aneurysm from Digital Subtraction Angiography (DSA) images. Our method relies on 3D U-net as the backbone and heavy data augmentation with a carefully chosen loss function. We were able to generalize well using our solution (despite training on a small dataset) that is demonstrated through accurate detection and segmentation on the test data.

Keywords: Aneurysm · Detection · Segmentation

1 Introduction

The leading cause of hemorrhagic stroke is the rupture of intracranial aneurysms. Aneurysms, in general, are vascular anomalies manifested as local *dilation* or balloon-like structure of blood vessels. Identifying intracranial aneurysm in the early stages of its development can reduce the risk of rupture and offer improved treatment planning. However, intracranial aneurysm detection is extremely challenging due to the variability in locations, shapes, and sizes. In clinical practice, different modalities are used for different stages of the diagnostic protocol. For preliminary screening, contrast-agent free modalities, such as TOF MRA or 3DRA, are the crucial and most commonly used modalities. Whereas images using contrast agents, such as DSA images, are used for advanced stages of treatment planning upon requirements.

Earlier approaches to detect cerebral aneurysms rely on 2D image processing, such as sphere enhancing filter [4]. It has been reported [2] that the convolutional neural network (CNN) based aneurysms detection method generalize better and can help radiologists to find more aneurysms without substantially decreasing their specificity. Deep learning-based methods can be classified into two categories, such as 1) global classification and 2) voxel-wise segmentation. While the former [6,16] is more memory and computation efficient, the latter is more useful

A. Hennemuth et al. (Eds.): CADA 2020, LNCS 12643, pp. 51–57, 2021.
https://doi.org/10.1007/978-3-030-72862-5_5

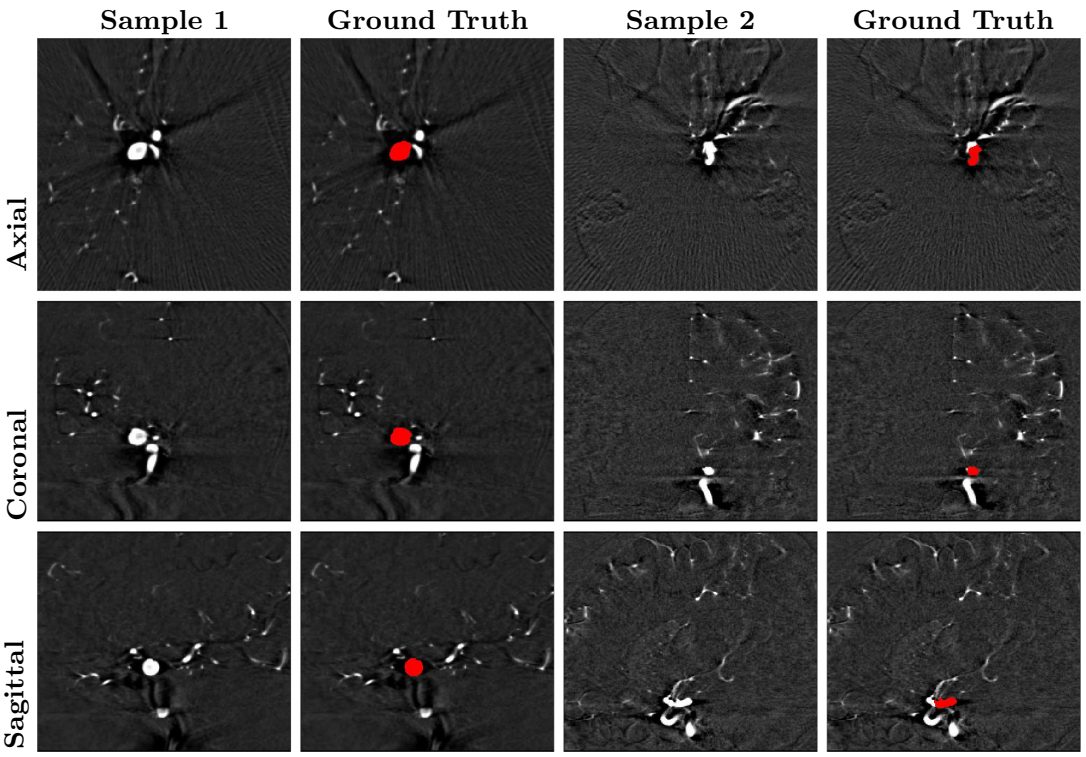

Fig. 1. Few training samples: The first, second, and third row shows the axial, sagittal, and coronal slice of the training examples, respectively. The odd columns show images, and the even columns show the corresponding ground truth annotation. Note the variability in the shape of the aneurysm to be detected by the network.

in providing the exact location and volume of the aneurysm. Several 2D U-net based [13] and 3D DeepMedic based [12] approaches were made to segment the aneurysms. The 3D network gives an extra performance gain at the cost of an increase in computational budget.

Detection and segmentation of cerebral aneurysm at an early stage is critical for clinical treatment planning. Digital subtraction angiography (DSA) is a commonly used modality to identify cerebrovascular pathology. An automatic algorithm to detect aneurysm from DSA images will accelerate the time requirements of the treatment pipeline. Keeping this in mind, we look for an effective solution to the CADA 2020 challenge. Some samples, along with their respective ground truth annotation, are presented in Fig. 1. We identify that the key features that separate aneurysm from a healthy vessel is the local blob-like structure, which can not be differentiated in 2D slices processing. Hence, we opt for a 3D approach. We also identify that given the amount of training samples, solving it as a pure object detection task would be difficult to learn [7]. Alternatively, the segmentation task is much easier to solve, and detection can be a simple post-processing stage. Given the number of aneurysms in a scan can vary, we rely on the assumption that no two aneurysms are adjacent to each other in voxel space and can be separated as different objects from the binary segmentation.

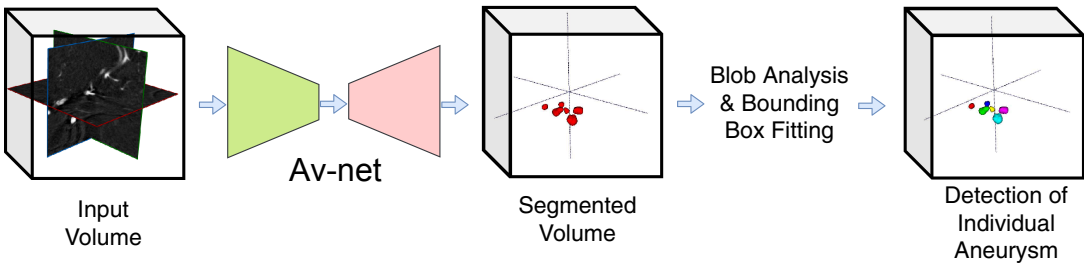

Fig. 2. Schematic overview of our proposed pipeline for detecting aneurysm from DSA images. The $128 \times 128 \times 128$ patch is processed with the $A\nu - net$ to produce the segmentation result. The segmentation result is processed to get rid of boundary artifacts. Subsequently, we do blob analysis and fit bounding boxes around each instance of segmented aneurysms to produce the individual aneurysm label.

Further, to enforce the network to learn shape-based discriminative features, we employ heavy data augmentation as described in Sec. 2.1. An overview of our proposed solution is presented in Fig. 2.

2 Methods

In this short paper, we describe our submission to the CADA 2020 challenge. We implement our segmentation as a three dimensional (3D) binary segmentation. After successful binary segmentation, we count the individual objects, and the final prediction considers one background class ("0") and "n" foreground classes depending on the number of objects in a 3D image volume.

The network architecture is inspired by the encoding-decoding architectures with skip connections. In other words, we use an architecture, which is very similar to the commonly used 3D U-Net architecture with some modifications [1]. We decide to use U-Net, because this is a very successful architecture for diverse medical imaging tasks [3,5,8,10,11,15]. We refer to our solution as Aν-net, which are homophones of 'Aneu-net' and 'an U-net.' The network architecture is depicted in Fig. 3.

Since Dice loss provides an edge at handling class imbalance [8,10] and cross-entropy loss is beneficial for smooth training convergence [9,14], we use both. The total loss function of our method is as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{Dice} + \mathcal{L}_{CrossEnt} \tag{1}$$

2.1 Implementation Details

Our encoder has four-stages with each stage having two residual blocks. We used instance normalization and parametric ReLU as the activation function. As a loss for the network, we used an equally weighted sum of the Dice Loss and the weighted cross entropy loss. The weights for the cross-entropy are [0.01 0.99] for the background and foreground, respectively. We did not use any additional

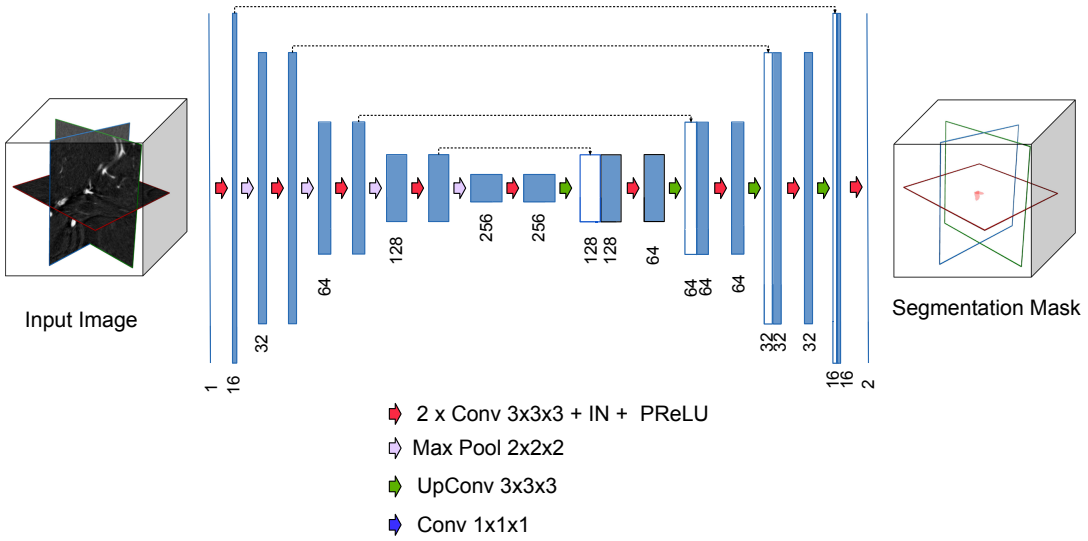

Fig. 3. Schematic overview of our proposed method: Our proposed *Aν-net* architecture is described in detail. The output of the final layer goes through a softmax layer before computing the loss function. For the dice loss, we exclude the background channel and only consider the foreground channel.

training data for this task. We used on the fly data augmentations such as random flipping in all three axes, random [90, 180, 270]-degree rotation along all three axes. We normalized the intensity value of each 3D scan to be of zero mean and unit standard deviation. All networks are implemented in Pytorch using the MONAI package. We use the Adam optimizer with a learning rate of 10^{-3}. The network was trained for 1000 epochs with a batch size of 2 and cubes of 128 voxels per dimension, keeping the configuration of the weight that performs best on the validation set, which split from the training set. The patches were sampled with a 4:1 ratio of the center of the patch being a foreground or background. These strategies help to alleviate the high-class imbalance in the data.

2.2 Detection from Segmentation

We post-process the predicted segmentation of the network to detect and fit the bounding box. We remove any segmentation which is smaller than 150 voxels. We select this threshold from the lower bound of the distribution of aneurysm size of the training data. We also remove artifacts that may appear in the boundary of the image by a simple masking. Subsequently, we do a blob analysis to identify the number of disconnected objects in the prediction. We thereby fit a rectangular bounding box to each instance of the segmented aneurysms.

3 Experiments and Results

We train on 100 scans and validate on 9 scans. We chose our model based on the best dice score on the validation set. We observe a dice score of 82.3 for our best

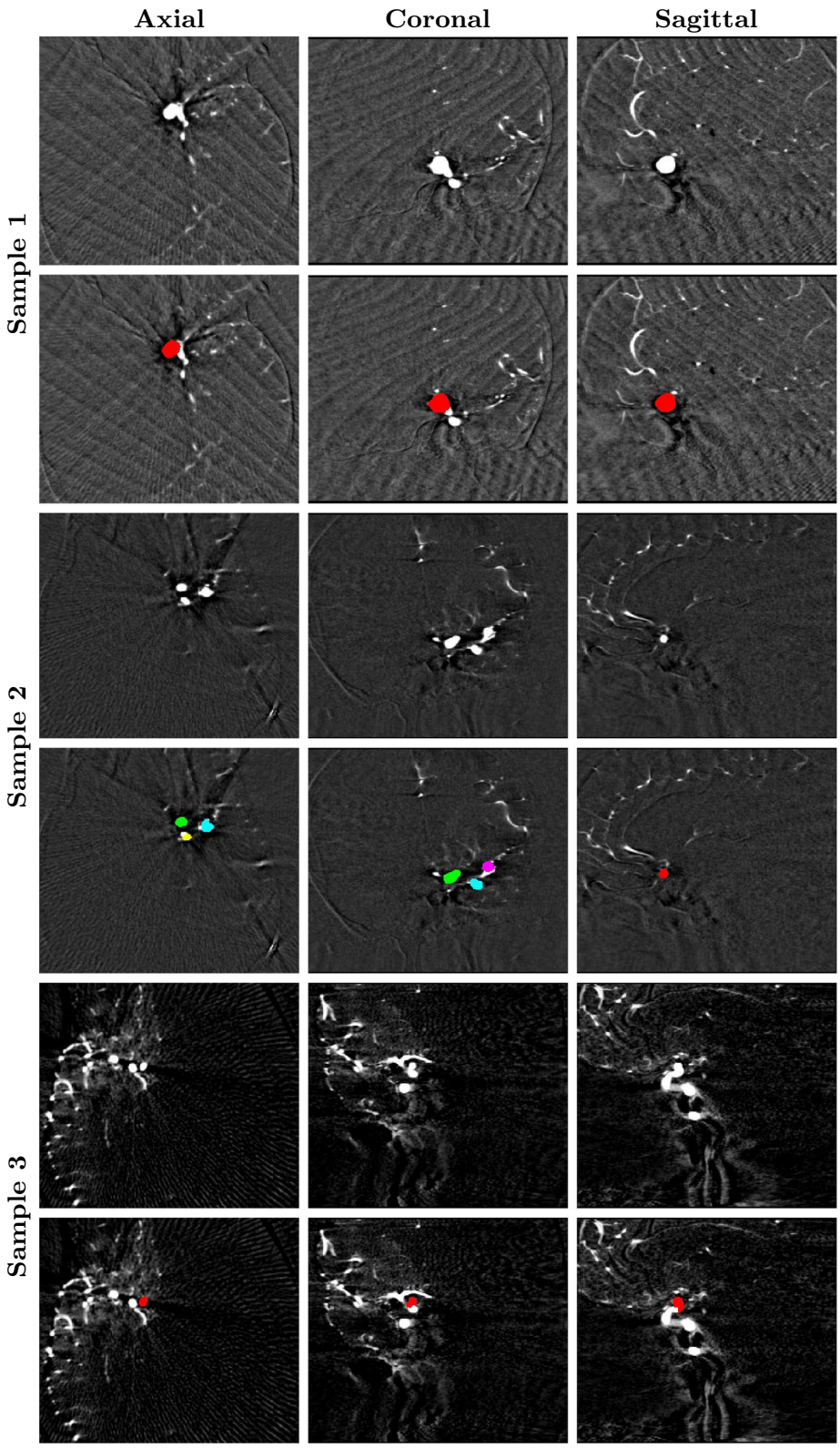

Fig. 4. Qualitative results: The first, second, and third columns represent the axial, sagittal, and coronal slice of the typical sample, respectively, from the test set. The odd rows denote the DSA images, and the even rows show the corresponding prediction from our method. We observe that our model effectively learns to detect multiple aneurysms at the same time.

model on the validation set. However, since the dice score largely varies based on the size of the aneurysm, we suspect that it may have produced a little lower dice score for the test set where most of the aneurysms were medium or small. We attribute this to the fact that during training we did not prioritize small aneurysms over the big ones. Figure 4 shows some qualitative results on the test dataset. Table 1 summarises our performance on the test dataset obtained from the official leaderboard.

Table 1. Our score on the test dataset under the team name IBBM as per the two leaderboards: https://cada.grand-challenge.org/evaluation/leaderboard/ and https://cada-as.grand-challenge.org/evaluation/leaderboard/.

Task	Detection	Segmentation
Score	0.8562	0.6817

4 Conclusions

We provide an effective solution for the CADA 2020 challenge using a simple U-net, without any additional training data and ensemble approach. We achieve accurate segmentation and detection results on all the test cases except a single case where our model does not detect any aneurysm. A minor drawback of our method is that it may struggle to differentiate multiple aneurysms located in one/two voxels' proximity to each other. Nonetheless, our proposed method can serve as a benchmark for developing more complex models aiming to better learn the discriminative anatomical features for aneurysm detection. Specifically, attention module can be used to improve performance on small aneurysms and distinguish aneurysms, which are close apart.

Acknowledgement. Suprosanna Shit and Ivan Ezhov are supported by the Translational Brain Imaging Training Network (TRABIT) under the European Union's 'Horizon 2020' research & innovation program (Grant agreement ID: 765148). Johannes C. Paetzold and Suprosanna Shit are supported by the Graduate School of Bioengineering, Technical University of Munich.

References

1. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 424–432. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_49
2. Duan, H., Huang, Y., Liu, L., Dai, H., Chen, L., Zhou, L.: Automatic detection on intracranial aneurysm from digital subtraction angiography with cascade convolutional neural networks. Biomed. Eng. Online **18**(1), 110 (2019)

3. Gerl, S., et al.: A distance-based loss for smooth and continuous skin layer segmentation in optoacoustic images. In: Martel, A.L., et al. (eds.) MICCAI 2020. LNCS, vol. 12266, pp. 309–319. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59725-2_30
4. Hentschke, C.M., Beuing, O., Paukisch, H., Scherlach, C., Skalej, M., Tönnies, K.D.: A system to detect cerebral aneurysms in multimodality angiographic data sets. Med. Phys. **41**(9), 091904 (2014)
5. Li, H., et al.: DiamondGAN: unified multi-modal generative adversarial networks for MRI sequences synthesis. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11767, pp. 795–803. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32251-9_87
6. Nakao, T., et al.: Deep neural network-based computer-assisted detection of cerebral aneurysms in MR angiography. J. Magn. Reson. Imaging **47**(4), 948–953 (2018)
7. Navarro, F., Sekuboyina, A., Waldmannstetter, D., Peeken, J.C., Combs, S.E., Menze, B.H.: Deep reinforcement learning for organ localization in CT. arXiv preprint arXiv:2005.04974 (2020)
8. Navarro, F., et al.: Shape-aware complementary-task learning for multi-organ segmentation. In: Suk, H.-I., Liu, M., Yan, P., Lian, C. (eds.) MLMI 2019. LNCS, vol. 11861, pp. 620–627. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32692-0_71
9. Paetzold, J.C., et al.: Transfer learning from synthetic data reduces need for labels to segment brain vasculature and neural pathways in 3D. In: International Conference on Medical Imaging with Deep Learning-Extended Abstract Track (2019)
10. Qasim, A.B., et al.: Red-GAN: attacking class imbalance via conditioned generation. Yet another medical imaging perspective. In: Medical Imaging with Deep Learning. PMLR (2020)
11. Shit, S., et al.: clDice–a topology-preserving loss function for tubular structure segmentation. arXiv preprint arXiv:2003.07311 (2020)
12. Sichtermann, T., Faron, A., Sijben, R., Teichert, N., Freiherr, J., Wiesmann, M.: Deep learning-based detection of intracranial aneurysms in 3D TOF-MRA. Am. J. Neuroradiol. **40**(1), 25–32 (2019)
13. Stember, J.N., et al.: Convolutional neural networks for the detection and measurement of cerebral aneurysms on magnetic resonance angiography. J. Digit. Imaging **32**(5), 808–815 (2019)
14. Tetteh, G., Efremov, V., Forkert, N.D., Schneider, M., Kirschke, J., et al.: Deepvesselnet: vessel segmentation, centerline prediction, and bifurcation detection in 3-D angiographic volumes. arXiv preprint arXiv:1803.09340 (2018)
15. Todorov, M.I., et al.: Machine learning analysis of whole mouse brain vasculature. Nat. Methods **17**(4), 442–449 (2020)
16. Ueda, D., et al.: Deep learning for MR angiography: automated detection of cerebral aneurysms. Radiology **290**(1), 187–194 (2019)