3D Coronary Artery Tree Reconstruction from Two Non-simultaneous Angiographic Projections using Deep Learning

Yiying Wang

Wolfson College

A thesis submitted for the degree of

Doctor of Philosophy

Supervised by

Prof. Vicente Grau and Dr. Abhirup Banerjee

Department of Engineering Science

University of Oxford

Trinity Term, 2025

Dedicated to my mom for her endless love

景秀濛汜，颖逸扶桑

Declaration

I declare that this thesis is entirely my own work and, except where stated, describes my own research.

Yiying Wang

Wolfson College

Acknowledgements

During my DPhil research journey, I am deeply indebted to my supervisors, Prof. Vicente Grau and Dr. Abhirup Banerjee, for their invaluable patience, unselfish support, and constructive feedback, which were instrumental in shaping my research trajectory. I also could not have undertaken this journey without the committee (Prof. Alison Noble, Prof. Jens Rittscher, and Dr. Anshul Thakur) for my Transfer and Confirmation of Status, who generously provided expertise and useful suggestions. I am very grateful to Prof. Alison Noble and Prof. Danail Stoyanov for their time serving on my defence committee; their critical perspectives and advice have been important in the finalisation of this thesis. I would be remiss in not mentioning my department and Wolfson college, as well as the WACV committee, which funded me to attend the conference in Tucson, where I had a most wonderful and enjoyable time.

I had the pleasure of working with cohort members in the Vicente's group and MultiMeDIA Lab (Atwany, Chen, Emmanuel, Guang, Haobo, Haorui, Jianing, Jingkun, Lei, Mingze, Mojtaba, Nicharee, Ning, Siyu, Thalia, Yuling, Yuxuan, and Zhengda), who inspired me. I am also thankful to have other nice friends (Ziyun, Chang, Jinquan, and Zhongyuan) and mentors (Prof. Qiu, Dr. Yang, and Dr. Huang), who helped and encouraged me.

Words cannot show my gratitude to my childhood friends (Kanghui, Pinjun, Qi, Juan, Shenghao, Shiqi, Yilian, Yuli, and Ruwen), who built a starry fire to spark up the darkest night of my life and keep me warm. How evergreen, our group of friends. I hope you shine and all the luck follows you around. I would also like to express my affection for the lovely black and tabby cats near my accommodation and the adorable dog, Guoba, for all the emotional values they contributed. My special thanks are extended to Taylor Swift, whom I have admired for nearly thirteen years. Attending the Eras Tour in London twice is among the most splendid moments of my life, which has soothed my soul.

Lastly, this endeavour would not have been possible without the sincere company of my beloved families (Guowei, Xiaohong, Yanlin, Guorun, Lewen, Lewu, Ziyun, and Kerou), who have often brought me comfort. Your unwavering belief in me has fuelled my spirits and motivation during the research. I wish you all happy and healthy every day. I feel most grateful to my mom, Guohui Chen, the greatest and best person in the universe, who means the world to me. I want to express my deepest thanks for the unconditional love you have shown, the sacrifices you have made, and everything you do for me. Your endless support and generosity have no bounds, even I want to touch the sky or reach the stars. By your side, I do not need to think about why all the trees change in the fall. Without you, I could never have made it this far to the Oxford. This thesis is in memory of all the mountains we moved and the walls we crashed through. I am so glad you are my mom and very lucky to have you in my life. You are my safe place and I know you are always on my side, no matter the distance. I want to acknowledge my own growth throughout this challenging journey. My old self was not particularly resilient and strong in the face of adversity, yet has now overcome life's greatest trials and all my flowers are growing back, though still occasionally overwhelmed with sorrow. You can shine in the dark and you know you are good when you can even do it with a broken heart. This thesis was the very first page, not where the story line ends. More life chapters are waiting for me to weave.

There were pages turned with the bridges burned

Everything you lose is a step you take

Make the friendship bracelets, take the moment and taste it

You've got no reason to be afraid

You're on your own, kid

You can face this

You're on your own, kid

You always have been

by $\mathcal{T}$aylor $\mathcal{S}$wift

This DPhil thesis was created using the LaTeX typesetting system.

Abstract

3D reconstruction of coronary artery trees from invasive X-ray coronary angiography (ICA) remains challenging due to the limited number of available projections and the temporal mismatch between clinically acquired views, which introduces motion-related inconsistencies. This thesis aims to develop automated deep learning-based approaches for reconstructing 3D coronary artery trees from only two clinical ICA projections acquired from a dual-axis rotational single-plane C-arm system, while accounting for realistic acquisition constraints. By reducing reliance on multiple projections and manual interpretation, such approaches have the potential to lower radiation and contrast exposure and improve consistency in the assessment of coronary anatomy.

This thesis first shows that 3D coronary artery trees can be reconstructed from only two projections without requiring explicit 3D supervision, by leveraging self-supervised neural representations, demonstrating the feasibility of sparse-view reconstruction under idealised conditions without inter-projection motion. We then consider the more realistic setting of non-simultaneous clinical ICA projections and train on projections simulated from coronary computed tomography angiography data with added rigid transformations, enabling the model to handle inter-projection motion implicitly and generalise to real clinical acquisitions. Building on this, we introduce an explicit iterative motion correction framework that compensates for misalignment between projections via registration in the 2D projection plane, leading to more stable and robust reconstruction across a range of motion levels representative of clinical acquisitions.

Finally, we investigate the impact of representation on reconstruction and show that replacing voxel-based formulations with point cloud representations provides a more efficient and flexible approach for modelling sparse, complex coronary artery structures, enabling higher-resolution reconstruction with reduced computational cost. Overall, this thesis demonstrates that accurate and anatomically consistent 3D coronary artery tree

reconstruction from only two clinically acquired projections is achievable under realistic imaging conditions, providing a principled learning-based framework with potential to reduce radiation and contrast exposure and improve consistency in clinical angiographic assessment.

List of Publications

The following is a list of articles that were published as a result of the research carried out for this thesis.

Wang, Y., Banerjee, A., Choudhury, R., & Grau, V. "DeepCA: Deep Learning-based 3D Coronary Artery Tree Reconstruction from Two 2D Non-simultaneous X-ray Angiography Projections". *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2025*. (**Oral: Top 8%**) (`https://arxiv.org/abs/2407.14616`)

Wang, Y., Banerjee, A., & Grau, V. (2024) "NeCA: 3D Coronary Artery Tree Reconstruction from Two 2D Projections via Neural Implicit Representation". *Bioengineering*, 11(12), 1227. (`https://arxiv.org/abs/2409.04596`)

Table of Contents

List of Figures

List of Tables

List of Acronyms

Clinics

CVDs	Cardiovascular Diseases
PCI	Percutaneous Coronary Interventions
FFR	Fractional Flow Reserve

Medical Imaging

CT	Computed Tomography
CCTA	Coronary Computed Tomography Angiography
PET	Positron Emission Tomography
SPECT	Single-photon Emission Computed Tomography
OCT	Optical Coherence Tomography
ICA	Invasive X-ray Coronary Angiography
DRR	Digitally Reconstructed Radiograph
DSA	Digital Subtraction Angiography
DARCA	Dual-axis Rotational Coronary Angiography
MRI	Magnetic Resonance Imaging
MRA	Magnetic Resonance Angiography
IVUS	Intravascular Ultrasound
ECG	Electrocardiogram

SE Stress Echocardiography

C-arm Geometry

CRA Cranial

CAU Caudal

AP Anterior-posterior

LAO Left Anterior Oblique

RAO Right Anterior Oblique

DSD Distance for Source to Detector

DSO Distance for Source to Origin

Anatomy of Coronary Arteries

RCA Right Coronary Artery

LCA Left Coronary Artery

LAD Left Anterior Descending

LCx Left Circumflex

RI Ramus Intermedius

Traditional Methods

FBP Filtered Backprojection

FDK Feldkamp-Davis-Kress

TV Total Variation

NURBS Non-uniform Rational Basis Splines

ICP	Iterative Closest Point

Deep Learning Methods

CNN	Convolutional Neural Network
DSConv	Dynamic Snake Convolution
ResNet	Residual Neural Network
DenseNet	Dense Convolutional Network
VAE	Variational Autoencoder
GAN	Generative Adversarial Network
WGAN	Wasserstein Generative Adversarial Network
WCGAN	Wasserstein Conditional Generative Adversarial Network
WCGAN-GP	Wasserstein Conditional Generative Adversarial Network with Gradient Penalty
DDPM	Denoising Diffusion Probabilistic Model
LDM	Latent Diffusion Model
NeRF	Neural Radiance Field
NAF	Neural Attenuation Field
3DGS	3D Gaussian Splatting
VGGT	Visual Geometry Grounded Transformer
ViT	Vision Transformer
CTL	Convolutional Transformer Layer
LLM	Large Language Model
VLM	Vision-language Model

FC	Fully Connected Layer
MLP	Multilayer Perceptron
RNN	Recurrent Neural Network
3D-R2N2	3D Recurrent Reconstruction Neural Network
LSTM	Long Short-term Memory
PCN	Point Completion Network
SPD	Snowflake Point Deconvolution
SPT	Snowflake Point Transformer

Evaluation

$\boldsymbol{Ot}(d)$	Overlap using a Sweeping Distance Threshold
$\boldsymbol{CD}_{\ell_2}$	Chamfer ℓ_2 Distance
EMD	Earth Mover's Distance
SDF	Signed Distance Function
Dice	Dice Similarity Coefficient
clDice	Centreline Dice Score
IoU	Intersection Over Union
reError	Reconstruction Error
MSE	Mean Squared Error
reMSE	Reconstruction Mean Squared Error
ASO	Almost Stochastic Order

Operators for Inverse Problems

ODL Operator Discretization Library

ASTRA All Scale Tomographic Reconstruction Antwerp

TIGRE Tomographic Iterative GPU-based Reconstruction

Others

SOTA State-of-the-art

GPU Graphics Processing Unit

XCAT Extended Cardiac-torso

Chapter 1

Introduction

1.1 Problem and Motivation

The term cardiovascular diseases (CVDs) comprises a group of disorders of the heart and blood vessels. CVDs are the most common cause of death in the US [1] and Europe [2]. In 2022, there are an estimated 19.8 million people who died from CVDs, representing 32% of all global deaths [3]. Approximately 80% of the world's deaths from CVDs take place in low- and middle-income countries [3]. Evidence [3] shows that CVDs and other non-communicable diseases lead to poverty at the household level due to high out-of-pocket expenses and catastrophic health spending. CVDs significantly strain the economies of low and middle-income nations on a macroeconomic basis. Thus, CVDs pose some of the most important questions in medical research. Early CVDs detection is fundamental as it allows immediate treatment or management, which could be crucial in determining the severity of the disease. Early CVDs identification, effective prognostic markers, and the available options for minimally invasive CVDs therapy have all driven the development in diagnostic and interventional imaging modalities for the anatomical and functional quantification of coronary arteries [4].

Cardiac catheterisation is an invasive interventional setup. Typically, it involves taking X-rays of the heart's arteries, or coronary arteries, using a technique called coronary angiography or arteriography. The resulting images are called coronary angiograms or arteriograms, which can provide important information about the function of the heart and the surrounding blood vessels supplying it. 2D invasive X-ray coronary angiography (ICA) is the gold standard during real-time cardiac interventional procedures and remains the most widely adopted imaging modality for CVDs assessment. There are around $250,000$

coronary angiograms performed across the UK every year [5]. Cardiac catheterisation and coronary angiography are usually safe [6], with some potential risks including uncommon allergic reactions to the contrast dye and bleeding under the skin where the catheter is inserted, which should stop after a few days. However, the 2D ICA projections generated from different angles of the 3D coronary artery tree make it difficult for cardiologists in clinical practice to mentally understand the global 3D coronary vascular structure. This is again complicated by the vessel overlap, foreshortening, complex vascular structure, and, most importantly, the artifacts caused by cardiac and respiratory motion and possible patient and device movements. These all may negatively affect the physicians' ability to assist in the navigation of interventional surgery in real-time [7]. Therefore, it is of great clinical significance to perform 3D coronary artery tree reconstruction based on 2D ICA projections.

A study [8] shows that computed tomography (CT)-associated cancer could eventually account for 5% of all new cancer diagnoses annually, if current utilisation practices and radiation dose levels persist. This demonstrates a need of radiation dose reduction, such as acquiring less projections or using low-dose CT. The potential chemotoxic adverse effects of a higher amount of radiographic contrast agent injected, higher radiation risk due to long-time exposure to X-rays, and longer procedural time due to percutaneous coronary interventions (PCI) limit the number of acquired angiographic projections: usually, only two angiographic projections are acquired per coronary artery. With such limited projections, a substantial amount of 3D coronary artery anatomical geometry information is lost, so the reconstructed images by traditional methods usually produce large artifacts and missing features. Even though it is possible to manually tackle these issues, it is time-consuming in processing and evaluation for each patient, as well as a cumbersome, repetitive and operator-dependent task [9]. Reducing intra- or inter-operator variability and enhancing the diagnostic quality with regard to time and accuracy could be achieved by an automated tool. Therefore, the development of an automatic 3D coronary artery tree reconstruction method based on limited (two) 2D ICA projections

is highly desirable for an objective, reproducible and relatively easier evaluation, as well as offering a novel and accurate approach to diagnosis with potential for clinical decision guidelines. A fully automated system could build 3D coronary artery tree during cardiac procedures, to reduce the risks of subjective interpretation of 3D coronary vasculature from 2D views with reduced foreshortening and vessel overlap, assist pre-operative planning (e.g., identifying the optimal projection angle for stent deployment, reducing unnecessary extra views, contrast agent volume, and radiation dose), give intra-operative direction and non-invasively measure physiological indices [4]. This would significantly ease the complexity of real-time diagnosis and interventional surgeries and will have considerable social and economic importance to teach and train inexperienced physicians, allowing comparatively complicated surgeries to be performed without the demand of comprehensive clinical experience [4].

Automated 3D coronary artery tree reconstruction from two ICA projections can significantly enhance the clinical assessment of coronary artery disease by providing a both anatomical and functional 3D model that overcomes the limitations of 2D projections. Fractional flow reserve (FFR) is an invasive, catheter-based technique used during angiography to measure pressure differences across a coronary artery stenosis, determining if a blockage restricts blood flow enough to require a stent. 3D models allow for computational fluid dynamics simulation to compute FFR without requiring an invasive pressure wire, so automated 3D coronary artery tree reconstruction can help functional evaluation of stenosis, often serving as a non-invasive alternative or adjunct to pressure wire-based FFR. Moreover, by providing accurate anatomical dimensions, automated 3D coronary artery tree reconstruction aids in accurately selecting the appropriate stent diameter and length, potentially reducing restenosis rates.

Recent advances in deep learning have reshaped task-specific problem-solving in various areas, including classification, segmentation, reconstruction, and more. Deep learning often outperforms traditional mathematical methods in both speed and automation. This thesis aims to implement automated 3D coronary artery tree reconstruction algorithms

based on deep learning techniques that can compensate for the non-rigid cardiac motion, using only two non-simultaneous ICA projections acquired from a dual-axis rotational single-plane C-arm imaging system.

1.2 Research Objectives and Contributions

The objective of this thesis is to provide automated 3D coronary artery tree reconstruction from only two non-simultaneous ICA projections, thus mitigating potential chemotoxic adverse reactions and radiation harm, as well as reducing the risks of subjective interpretation of 3D vasculature from 2D views. To achieve this goal, we first explored the feasibility of 3D coronary artery tree reconstruction from only two projections without any motion between projection planes. Next, we designed a pipeline to obtain 3D coronary artery tree reconstruction from two clinical non-simultaneous ICA projections. Based on this pipeline, we introduced an iterative registration strategy to correct motion between two non-simultaneous projections to refine 3D coronary artery tree reconstruction results. Finally, to overcome the issue of inefficient learning of sparse, complex coronary vessel structures based on voxel representation, we developed and validated an efficient point cloud-based approach for dense 3D coronary artery tree surface reconstruction. The outlines for each contribution are as follows.

Contribution 1: 3D Coronary Artery Tree Reconstruction from Two 2D Projections via Neural Implicit Representation

- Proposed a self-supervised learning method based on implicit neural representation that iteratively optimises 3D reconstruction results from two 2D projections where neither 3D ground truth for supervision nor large training datasets are required.
- Utilised the advantages of multiresolution hash encoder to allow efficient feature encoding, residual multilayer perceptrons as a continuous function to represent the

3D coronary artery tree, and differentiable cone-beam forward projector layer to simulate projections to enable self-supervised learning.

- Validated our model in coronary computed tomography angiography (CCTA) data of both right coronary artery (RCA) and left anterior descending (LAD) in terms of six effective metrics.

Contribution 2: 3D Coronary Artery Tree Reconstruction from Two 2D Non-simultaneous X-ray Angiographic Projections

- Proposed a novel deep learning pipeline to implicitly compensate for the non-rigid motion between two non-simultaneous ICA projections to achieve 3D coronary artery tree reconstruction. To the best of our knowledge, this is the first study using deep learning to achieve 3D coronary artery tree reconstruction from two clinical non-simultaneous ICA projections, providing a baseline for future improvement in this area.
- Leveraged the Wasserstein conditional generative adversarial network with gradient penalty, latent convolutional transformer layers, and a dynamic snake convolutional critic for training.
- Achieved generalisation to two non-simultaneous ICA projections through simulating 2D projections from 3D CCTA data with simulated motion on the second projection plane.
- Overcame the problems of both limited number of real paired ICA data with projection geometry information and unavailable 3D ground truth for real ICA data.
- Validated our model on a CCTA dataset and a clinical ICA dataset (unseen domain) with an application-specific evaluation method to tackle the deformation in 3D reconstructions, unavailability of 3D ground truth for real ICA scans, and motion between projection planes, together with Chamfer ℓ_2 distance.

Contribution 3: Deep Iterative Motion Compensation for 3D Coronary Artery Tree Reconstruction from Two 2D Non-simultaneous Projections

- Proposed a deep learning-based pipeline to iteratively correct rigid and non-rigid motions between two non-simultaneous projections to refine 3D coronary artery tree reconstruction results via 2D registration.
- Investigated the effect of simulated 2D projections with different misalignments from 3D CCTA data on the generalisation to real non-simultaneous ICA data.
- Validated our model based on both a CCTA dataset and three unseen datasets including a real ICA dataset, a synthetic dataset, and a different CCTA dataset from another country.

Contribution 4: Snowflake Point Transformer with Point Adversarial Loss for Coronary Tree Reconstruction from Two Non-simultaneous Projections

- Leveraged a point cloud-based representation to efficiently handle vessel topology to achieve dense point cloud-based coronary artery tree reconstruction.
- Compensated for the non-rigid motion between projections implicitly to facilitate reconstruction from two non-simultaneous projections, through simulating projections from CCTA point cloud data.
- Supported a dynamic size of different point clouds in one input batch, allowing various vasculature reconstructions.
- Proposed Wasserstein conditional point adversarial learning module with gradient penalty that is specifically designed for point cloud structures, allowing accurate distinguishing between reconstruction and ground truth point clouds.

Chapter 2

Background and Literature Review

2.1 Clinical Background

2.1.1 Cardiac Anatomy

The heart is a muscle, namely cardiac muscle or myocardium, about the size of the fist, in the middle of the chest tilted slightly to the left. It works like a pump that contracts and relaxes to circulate blood to deliver oxygen and nutrients throughout the body, helping the body, such as organs, function properly [10].

The heart has both left and right sides. Both sides have an upper chamber (atrium) that receives blood flowing into the heart and lower chamber (ventricle) that pumps blood out of the heart. More specifically, electric impulses stimulate the muscles, causing the sequential contraction of atria and ventricles. With the contraction of the myocardium sequentially compressing the heart's left and right chambers [11], pressure differentials caused by asynchronous contractions of the ventricles and atria cause blood to circulate through the body (deoxygenation) and the lungs (oxygenation) [12]. Several pathologies can be diagnosed using the electrical signals recorded on an electrocardiogram (ECG) [13, 14]. Additionally, during image acquisition, the ECG signal offers information about the heart phase, or motion state.

The blood supply to the myocardium comes from the left and right coronary arteries, which run on the epicardium, i.e., the surface of the heart. When the coronary arteries are narrowed by fatty material in their walls, coronary artery disease, or coronary heart disease, is developed. Figure 2.1 depicts the anatomy of the heart and most important coronary arteries.

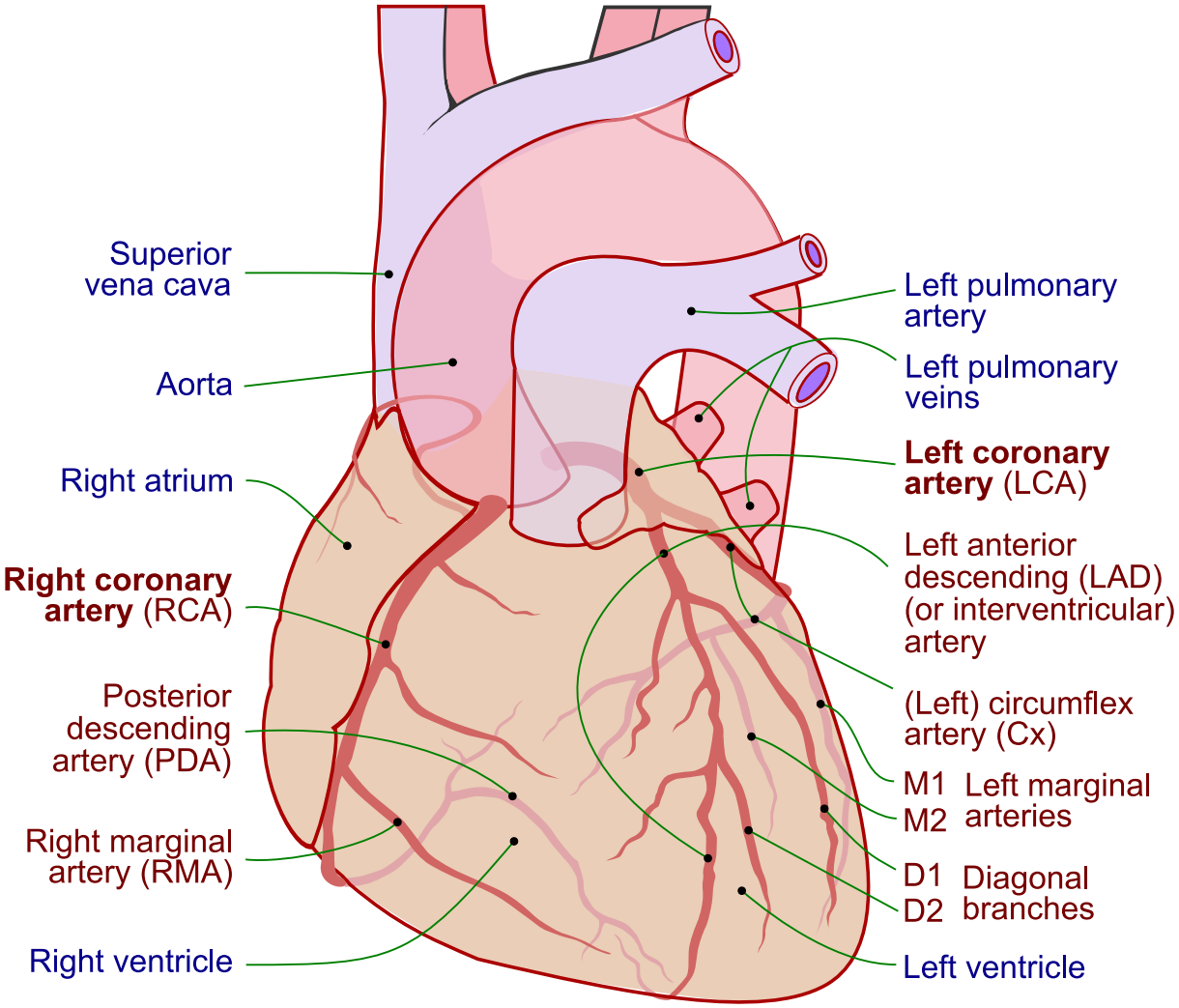


Figure 2.1: Anatomical diagram of the heart, showing the coronary arteries. (Taken from [15])

2.1.2 Clinical (cath. lab.) Setting

Cardiac catheterisation and coronary angiography [16, 17] are usually carried out in an X-ray room or a catheterisation laboratory at a hospital or specialist heart centre. During the cardiac catheterisation procedures, a long, thin, flexible tube (catheter) is inserted into a blood artery through the patient's arm or groin by a heart specialist (cardiologist) after cleansing with antiseptic fluid. The tip of the catheter is moved up the vessel until it reaches the heart and coronary arteries, using X-ray angiograms as a guide. Patients are connected to an ECG machine throughout the procedures to record their heart rhythms and the electrical signals of each heartbeat. After injecting a specialised dye, i.e., a primarily iodine-based contrast medium, via the catheter, a sequence of X-ray images of

the heart and the surrounding blood arteries, known as angiograms, is obtained. When the fluid flows, the blood vessels are revealed by the contrast medium on the angiograms. This makes it possible to identify any arteries that are narrowed or blocked. Although the area where the catheter is inserted is numbed to relieve pain, patients are awake during the process. If no further interventional procedures are needed, such as balloon angioplasty, the procedure should take about 30 minutes.

2.1.3 Invasive X-ray Coronary Angiography (ICA)

Current clinical decision making for the diagnosis and severity of CVDs relies heavily on the medical images acquired through different imaging modalities, either diagnostic, such as CCTA and magnetic resonance angiography (MRA), or interventional, such as ICA [4]. Regardless of the emergence of 3D non-invasive imaging modalities (CCTA, MRA) to visualise the coronary arteries, ICA remains as the most widely-used technique in clinical decision making and therapy guidance [18]. Coronary angiography can be used to help diagnose several heart conditions, including coronary heart disease, heart attacks, angina, congenital heart disease in children, valvular heart disease, and cardiomyopathy [19]. It is widely adopted for CVDs diagnosis [9], due to its advantages in available trained personnel [4], high spatial and temporal imaging resolution, usefulness during coronary interventions in real-time, and ability to provide both diagnostic information and guidance in following therapeutic procedures, in contrast to CCTA or MRA [20]. It is also thought to be the most effective way to diagnose coronary heart disease, in cases when the accumulation of fatty substances in the coronary arteries compromises the heart's blood flow [6].

All persons $\geq$ 18 years of age in England who had a cardiac imaging investigation within the National Health Service between 2012 and 2018 were included in a retrospective observational nation-level study [18], with individual follow-up continuing until the end of 2019. This study shows that ICA remained the commonest imaging modality performed with stable usage across years. According to the study, there is an average of 214.5 ICA angiograms scanned per $100,000$ population every year. Figure 2.2 illustrates England-

wide examinations performed per month per 100,000 population for the investigation of coronary artery disease from 2012 to 2018, in terms of different modalities, i.e., ICA, single-photon emission computed tomography (SPECT), CCTA, magnetic resonance imaging (MRI), stress echocardiography (SE), and positron emission tomography (PET). It shows an increased demand in CCTA, but ICA remains stable, still being the most widely used imaging service in 2019. ICA is irreplaceable during cardiac intervention. Recent data about the number of PCI which requires ICA performed in England and Wales across years [21] are illustrated in figure 2.3. It shows continuing stable amounts across years, except for a decrease during the COVID-19 pandemic, but with a subsequent recovery since moving out of the pandemic.

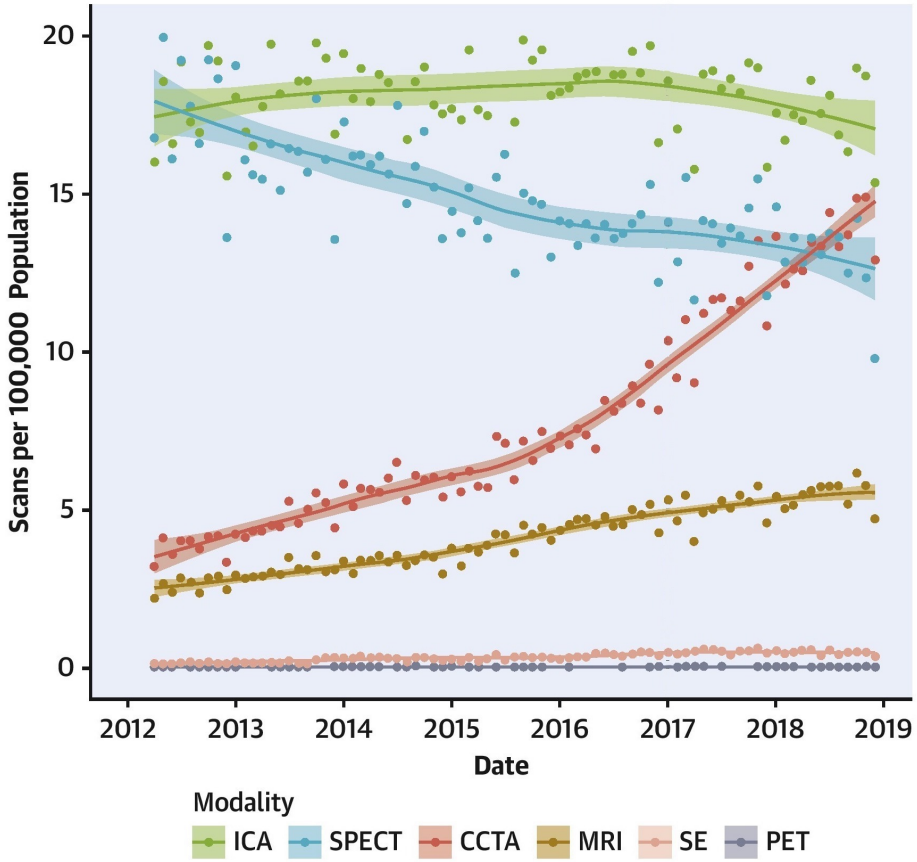


Figure 2.2: England-wide examinations performed per month per 100,000 population for the investigation of coronary artery disease from 2012 to 2018, in terms of different modalities, i.e., ICA, SPECT, CCTA, MRI, SE, and PET. (Adapted from National Trends in Coronary Artery Disease Imaging: Associations With Health Care Outcomes and Costs [18])

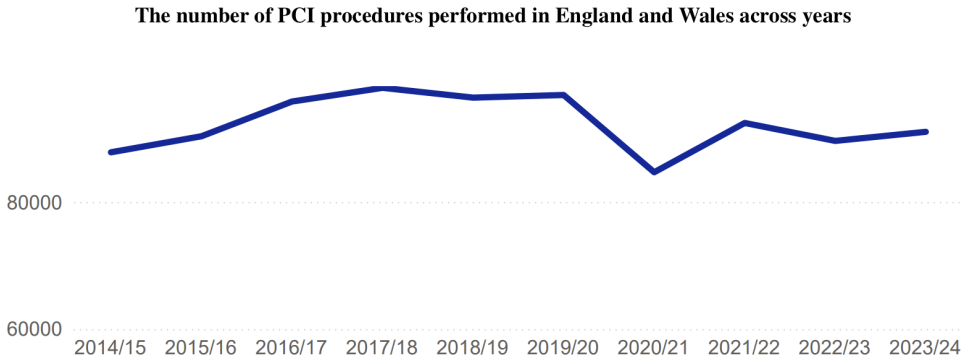


Figure 2.3: The number of PCI performed in England and Wales across years. (Adapted from National Audit of Percutaneous Coronary Intervention (NAPCI): 2025 Annual Report [21])

2.1.4 Imaging Geometry for ICA

Since ICA produces 2D angiographic projections of the complex 3D anatomy of coronary arterial tree, there is a need to collect multiple projections by putting the X-ray source and detector in specific directions for an easier CVDs evaluation. Positioning is controlled by the C-arm of the angiography imaging system, with specific distance settings, i.e., distance for source to detector (DSD) and distance for source to origin (DSO). The C-arm is a C-shaped imaging system, which contains an X-ray source and a flat panel detector at its end. According to different C-arm setups, movement of the source and detector along different axes is allowed. The capability of this movement is the main design parameter that distinguishes different types of ICA protocols [22]. Based on this capability, different types of ICA exist, namely single-plane (standard and conventional), bi-plane, rotational, and dual-axis rotational coronary angiography (DARCA) [4]. A standard single-plane X-ray angiography system can only collect images from a few fixed, operator-chosen views. A bi-plane system contains two C-arms and is generally configured to acquire simultaneous projections from orthogonal views. Rotational angiography enables the acquisition of a series of images from a predefined C-arm rotation [23], while DARCA enables the acquisitions with additional cranial (CRA) or caudal (CAU) angulation during one individual scan [24] that improves the safety of patients and simplifies the acquisition of angiographic projections.

On a dual-axis rotational single-plane C-arm system, a projection image is generated based

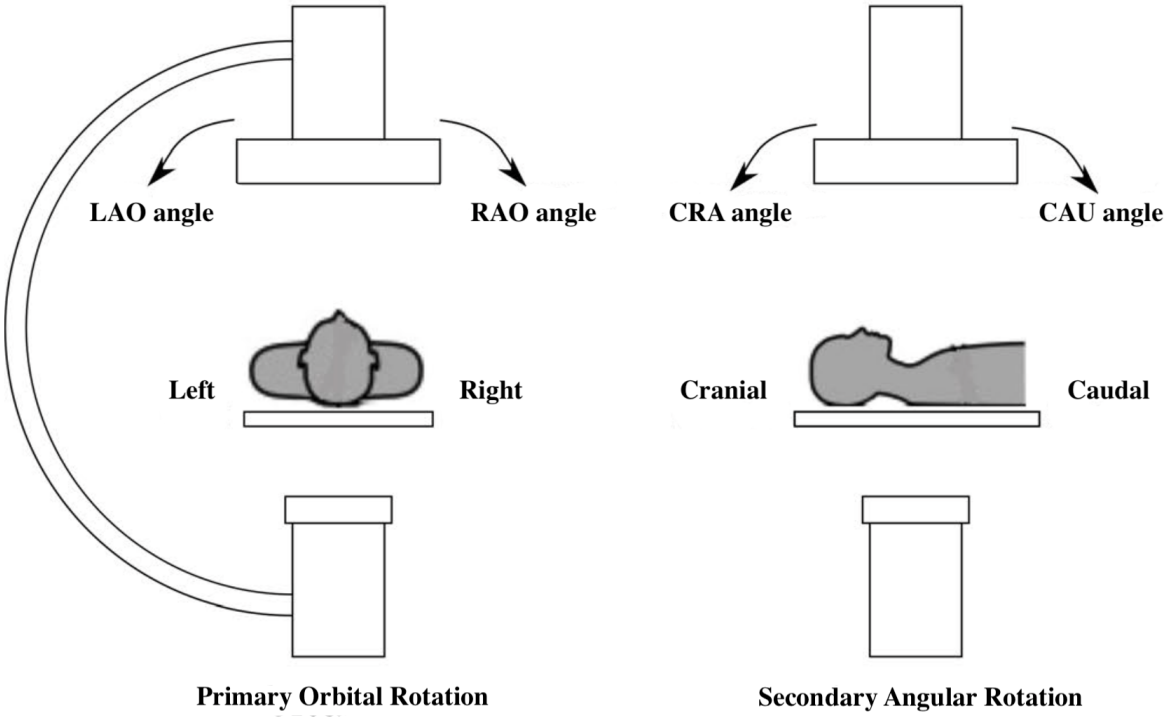


Figure 2.4: Dual-axis rotational C-arm imaging geometry. (Adapted from [25])

on both primary and secondary angles [25], as illustrated in figure 2.4. The primary angle is set based on orbital rotations, which are the rotations within the plane of the C-arm gantry. The value of the primary angle is 0 when the rotation is exactly at anterior-posterior (AP) position. The left anterior oblique (LAO) view implies a primary angle larger than 0, while the right anterior oblique (RAO) view implies a primary angle lower than 0. The secondary angle is set from angular rotations, which are the rotations perpendicular to the plane of the C-arm gantry. Its value is 0 when the rotation is at a straight position, larger than 0 when rotating towards the CRA angulation, and less than 0 when rotating towards the CAU. This C-arm setup enables image acquisition of a patient from any projection angles, only restricted by the anatomy of that patient. Therefore, it is possible to perform a scan for both RCA and left coronary artery (LCA) by adjusting the position angles. However, in single-plane systems, projection images are generated non-simultaneously and thus are prone to motion artifacts, including cardiac motion, respiratory motion, patient or imaging device-related movements, etc. All types of angiography systems acquire 2D projections of coronary arteries on an image plane (flat-detector) over time.

2.2 Deep Learning for 3D Reconstruction from a Small Number of Views

Reconstructing from limited views is an ill-posed problem, and recent deep learning algorithms have shown promising performance on this task. In this section, we will first introduce some deep learning fundamentals. Then, different deep learning methods for general object/scene reconstruction as well as medical image reconstruction will be discussed.

2.2.1 Deep Learning Fundamentals

There were two main historical waves of research on artificial neural networks [26]. The first wave began with cybernetics in the $1940s$–$1960s$, when theories of biological learning were developed [27, 28] and the first models, like the perceptron [29], which allowed for the training of a single neuron, were implemented. The second wave began with the connectionist method of the $1980-1995$ era, which used backpropagation to train a neural network with one or two hidden layers [30]. In 2006, Hinton *et al.* [31] demonstrated that a type of neural network, known as deep belief network, could be effectively trained using a technique known as greedy layer-wise pretraining. This discovery marked the beginning of the current and third wave of neural networks research [31–33]. In order to highlight the fact that researchers could now train deeper neural networks than previously conceivable and to draw attention to the theoretical significance of depth, this wave of neural networks research popularised the term "deep learning" [26].

Thanks to deep learning [34], intricate structures in high-dimensional data can be found with relative ease, which enables computational models made up of many processing layers to learn representations of data with different levels of abstraction. In order to specify how a machine should modify its internal parameters, which are required to calculate the representation in each layer based on the representation in the preceding layer, it employs the backpropagation technique. In the following subsections, we will discuss

several essential deep learning milestone models which have reshaped the pattern of many domains.

Multilayer Perceptron (MLP)

A fully connected (FC) layer (dense layer) is a fundamental building block of neural networks. Every neuron in a FC layer is fully connected to every neuron in the next layer, as illustrated in figure 2.5. An multilayer perceptron (MLP) is a specific type of feedforward neural network architecture that uses FC layers. It consists of an input layer, one or more hidden layers which are often FC layers, and an output layer, using non-linear activation functions to learn complex patterns and solve problems that are not linearly separable. In an MLP, each neuron computes a weighted sum of its inputs and adds a bias term, and this value is passed through a nonlinear activation function to introduce nonlinearity, which is essential for learning complex functions. Fully connected MLP layers have been used at the last layers of many milestone deep learning models to map learned features to specific classifications, such as LeNet [35], AlexNet [36], residual neural network (ResNet) [37], and VGGNet [38], and as the backbone in Transformers [39]. Moreover, MLPs can approximate any continuous function given sufficient width and depth. This property has been leveraged in neural radiance field (NeRF) [40] where MLPs are used to represent the underlying continuous volumetric scene function.

Convolutional Neural Network (CNN)

The filter used in the convolutional layers is called a convolution kernel, which is a small matrix sliding over the input data in a process called convolution. As illustrated in figure 2.6, the convolution operation performs element-wise multiplication between the kernel and a region of the input, followed by summing the results into a single value, which is repeated across the entire input. Convolution kernels are learned during training to detect more complex and abstract features as the network deepens. By using multiple kernels, convolutional neural network (CNN) can automatically learn a wide range of

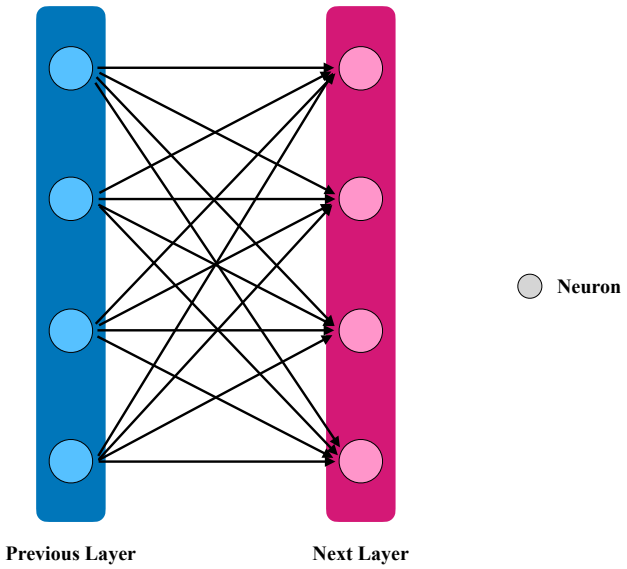


Figure 2.5: An illustration of a fully connected layer or block.

features at various levels of abstraction without manual feature extraction. The size and depth of the kernels are important hyperparameters, influencing how well the network captures the spatial relationship between pixels. LeCun *et al.* [35, 41] proposed and implemented the first practical CNN, called LeNet, which has successfully been applied to handwritten zip code recognition. It uses convolutional layers to detect spatial features, subsampling (pooling) layers to reduce dimensionality, and shares weights to reduce the number of parameters. The whole recognition process, from the character's normalised picture to the final classification, is learned by a single network via backpropagation. The Modified National Institute of Standards and Technology database (MNIST) of handwritten digits [42] was then created for recognition evaluation, and it showed CNNs outperform all other models [43].

In the past decade or so, advances in hardware with significantly faster computations have enabled researchers to scale up their models quickly, and the availability of massive datasets has allowed deep learning models to generalise better and learn more complex patterns. A large-scale hierarchical image database, ImageNet [45], with tens of millions of annotated images, was introduced, which offers unparalleled opportunities to researchers in the computer vision community. Using ImageNet, a deep convolutional network, AlexNet [36]

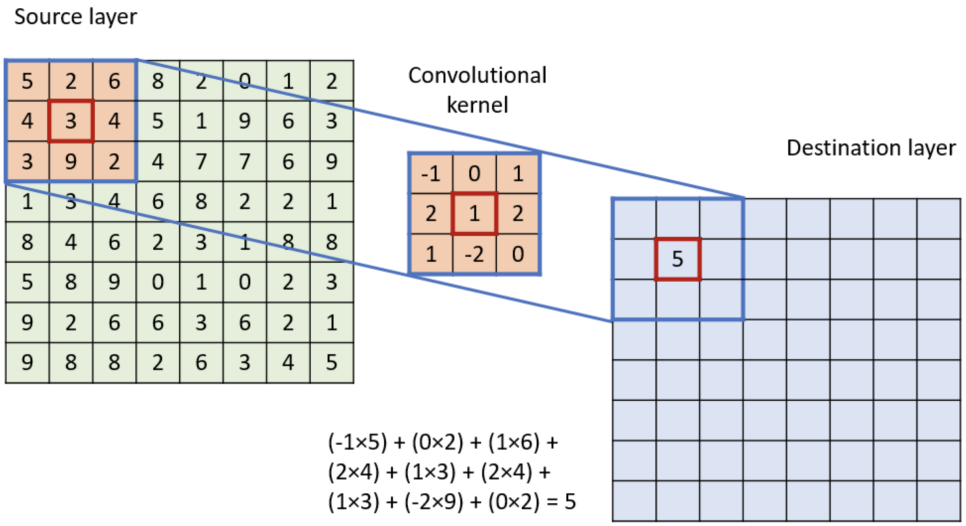


Figure 2.6: A representation of a convolution process. The convolutional kernel moves over the source layer to fill the pixels in the destination layer. (Taken from [44])

brought breakthroughs in the task of image classification for $1,000$ different classes, which marked the beginning of deep learning revolution in computer vision. AlexNet consists of 5 convolutional layers and 3 fully-connected layers, which have 60 million parameters and $650,000$ neurons. It provided a highly optimised, efficient Graphics Processing Unit (GPU) implementation of the 2D convolution operation. Later, VGGNets [38] showed performance improved with very deep convolutional networks on ImageNet by pushing the depth to $16 - 19$ layers.

However, simply increasing the network depth to capture more complex features could cause vanishing gradients and performance degradation. He *et al.* [37] proposed a deep residual learning framework with residual connections, ResNet, to solve this problem. Residual connections in feedforward neural networks are shortcut connections that skip one or more layers. These connections only carry out identity mapping in ResNet, and the outputs of these connections are appended to the outputs of the stacked layers. There is no additional computational complexity or parameter added by the identity shortcut connections. ResNet makes optimisation easy and demonstrated substantial accuracy gains in many tasks, including ImageNet classification, even with greatly increased depth of layers ($8\times$ deeper than VGGNets [38]). Dense convolutional network (DenseNet) [46] is another deep convolutional architecture that alleviates the vanishing-gradient problem

and obtained significant improvements in many tasks, including ImageNet classification. In DenseNet, each layer uses its own feature maps as inputs into all subsequent layers. DenseNet concatenates features instead of combining them like ResNet does, which sums features before passing them into a layer. Because it promotes feature reuse, it uses fewer parameters than conventional convolutional networks, which aids in the training of deeper networks. To enable training with high-resolution images under limited GPU resources, U-Net [47] was proposed, demonstrating fast and superior performance in the task of medical semantic segmentation. U-Net consists of a usual contracting path to capture context and a symmetric expanding path where pooling operators are replaced by upsampling operators to increase the resolution of the output, yielding a U-shaped architecture. For precise localisation, 'copy and crop' skip connections are applied where high-resolution features from the contracting path are combined with the upsampled output.

By combining channel-wise and spatial information inside local receptive fields at each layer, the convolution operator allows networks to create useful features. Hu *et al.* [48] suggested a block that explicitly models channel interdependencies to adaptively recalibrate channel-wise feature responses, which showed significant improvements in image classification. Moreover, due to the fixed geometric structures in traditional convolution operations, Dai *et al.* [49] proposed deformable convolution and deformable region-of-interest pooling to enhance their transformation modelling ability by adding more offsets to the spatial sampling locations and learning the offsets from the target tasks without extra guidance. Qi *et al.* [50] proposed dynamic snake convolutional (DSConv) to capture the characteristics of tubular structures accurately by adaptively concentrating on local structures that are thin and tortuous.

Generative Models

A generative model is a kind of machine learning model that creates new data samples which are similar to the original data by learning the dataset's underlying distribution. It focuses

on modelling the joint probability distribution of both the inputs and outputs, essentially learning how the data is structured and generated. Variational autoencoder (VAE) [51] was proposed to learn a probabilistic mapping between a latent space and data, allowing new data generation by sampling from this space. It introduced a variational inference technique to approximate the posterior distribution over the latent variables, which allows for efficient learning of complex, high-dimensional data distributions, and the reparameterisation trick of the variational lower bound that allows for efficient backpropagation through stochastic variables. A simple FC neural network MLP was initially used in VAE [51] and then as CNN became popular, convolutional layers were integrated in VAE [52] for generating high-quality images where deconvolutional layers (transposed convolutions) were applied in the decoder to reconstruct images from the latent variables. VAEs are easy to train and capable of capturing the entire data distribution, but can sometimes produce blurry results, sacrificing fine details in favour of reducing pixel-wise reconstruction errors.

A generative adversarial network (GAN) [53] was introduced that excels at high-fidelity generation. A GAN model consists of two neural networks, a generator and a discriminator, which compete against each other simulating a minmax two-player game. The generator captures the data distribution to create synthetic data, while the discriminator estimates the probability of how real or fake the generated data is, based on the training data. During training, the generator aims to improve its ability to create data that can fool the discriminator, while the discriminator seeks to become better at distinguishing real from fake data. Over time, this adversarial process leads to the generation of highly realistic samples. Based on GANs [53], condition GANs [54] were proposed to generate data conditioned on specific inputs such as label or attributes and deep convolutional GANs [55] were introduced to learn a hierarchy of representations in the discriminator and generator, ranging from scenes to object pieces.

The traditional GAN model often suffers from unstable training problems [56, 57] such as difficulty in convergence, vanishing gradient, and mode collapse, where the generator might be happy enough to produce only a small part of the data diversity. Wasserstein generative

adversarial network (WGAN) [58] improved these issues by introducing a different loss function based on Wasserstein distance (Earth Mover's distance). WGAN sometimes can still generate only poor samples or fail to converge. Further to this, Gulrajani *et al.* [59] proposed a gradient penalty to penalise the norm of the gradient of the critic with respect to its input to enable stable training of a wide variety of GAN architectures with almost no hyper-parameter fine-tuning. The architecture of GANs also brought novel applications such as style transfer. In order to solve image-to-image translation difficulties, Isola *et al.* [60] explored conditional adversarial networks, which are useful for colorising pictures, reconstructing objects from edge maps, and synthesising photographs from label maps. The training sets of aligned image pairs are required to learn this image-to-image mapping, but they are usually unavailable for many tasks. CycleGANs [61] explored unpaired image-to-image translation using cycle-consistent adversarial networks, which can translate images from one domain to another without paired data, such as turning a horse image into a zebra. CycleGANs include two basic GAN structures, one for direct mapping and another one for inverse mapping as a constraint. A cycle consistency loss was introduced to ensure the model can translate an image from one domain to another and then back to the original domain, resulting in the original image. This forces the generator to learn to preserve the underlying content of the image while changing its style.

A denoising diffusion probabilistic model (DDPM) [62] was proposed recently, producing high-quality image synthesis while having the advantages of easy and stable training as well as the ability to capture the whole data distribution, compared to GANs. DDPMs introduce noise to a sample gradually over several steps until it becomes pure noise, and then learn how to reverse this process to reconstruct the data from noise. The sampling procedure of diffusion models is a type of progressive decoding. Based on DDPMs, Song *et al.* [63] used score-based matching techniques to optimise the diffusion process and Dhariwal and Nichol [64] used a diffusion model to generate images based on specific conditions. However, DDPM-based models are computationally expensive due to the direct operations in pixel space, and the inference is expensive due to sequential evaluations. A

latent diffusion model (LDM) [65] was proposed that operates in a compressed, lower-dimensional latent space to generate images, which achieved a nearly ideal balance between preserving details and reducing complexity, significantly increasing visual quality. LDMs are flexible and achieved superior performance in various tasks such as text-to-image synthesis while dramatically reducing computational requirements compared to DDPMs.

The schematic illustration of the comparisons between the three aforementioned generative models, i.e., VAEs, GANs, and diffusion models, is presented in figure 2.7.

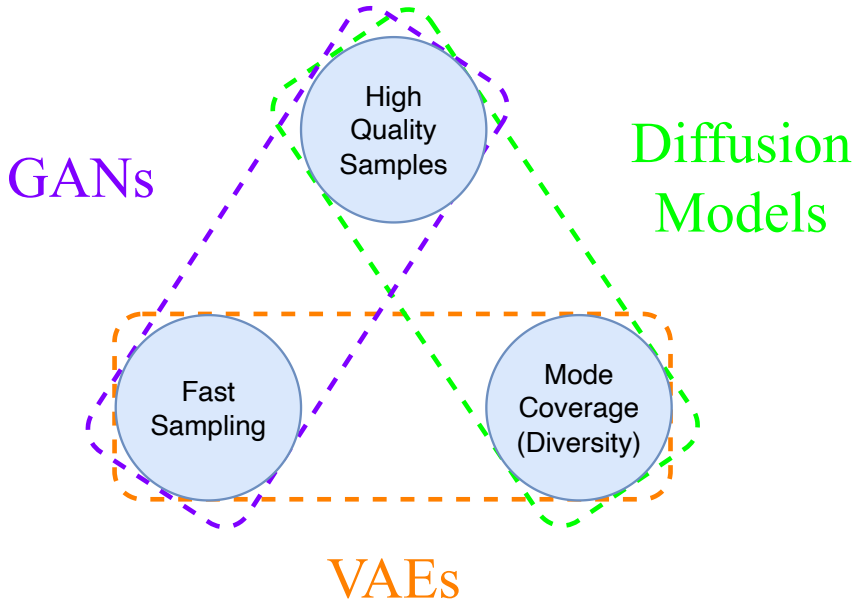


Figure 2.7: Comparisons between three main generative models, i.e., VAEs, GANs, and diffusion models.

Transformers

Transformer [39] is based solely on attention mechanisms [66], dispensing with recurrence and convolutions entirely. It is designed to handle sequential data that can capture long-range dependencies more efficiently and in parallel, significantly improving performance on natural language processing tasks. Adapting the vanilla Transformer architecture, Vision Transformer (ViT) [67] was proposed for computer vision tasks, as illustrated in figure 2.8. ViT divides an image into non-overlapping fixed-size patches, which are then linearly embedded and processed in a Transformer framework to capture global dependencies

across the image. The self-attention mechanism of Transformers is directly applied to image patches, treating them as sequences of tokens similar to words in a sentence. ViT demonstrated that Transformers could outperform traditional CNNs, particularly with large datasets and sufficient computational resources. Following the success of ViT, in order to increase scalability, Swin Transformer [68] created hierarchical vision transformers whose representation is calculated using shifted windows. By restricting self-attention computation to non-overlapping local windows and permitting cross-window connectivity, the shifted windowing technique increases efficiency. He *et al.* [69] proposed an efficient and effective masked autoencoder where ViT attends to a set of unmasked patches and a smaller decoder tries to reconstruct the pixel values of random masked patches. This scalable approach allows for learning high-capacity models that generalise well. A highly generalisable foundation segmentation model, Segment Anything Model (SAM), was introduced [70] based on Transformer architectures [39, 67, 69], which demonstrated impressive zero-shot performance. Transformer has now become a foundational model for vision tasks like image classification, segmentation, and object detection.

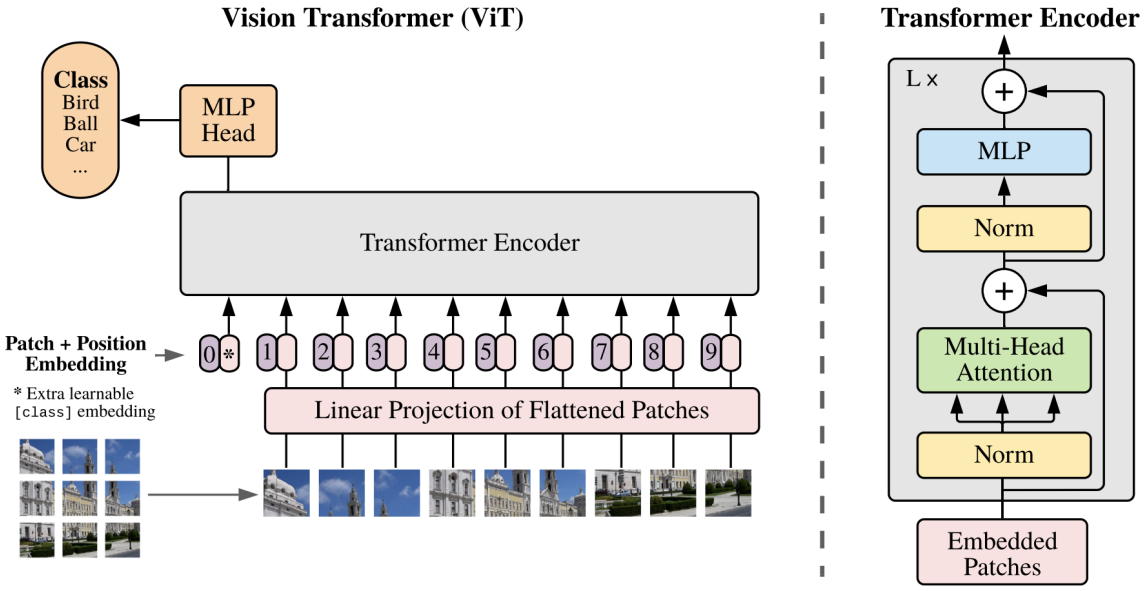


Figure 2.8: ViT model overview. An image is split into fixed-size patches, each of which is linearly embedded, added with position embeddings, and the resulting sequence of vectors is fed to a standard transformer encoder. (taken from [67])

Point Cloud-based Neural Networks

CNNs are primarily designed for processing grid data such as 2D/3D images, and the computational cost of CNNs limits the resolution of grid data. Point cloud representation of 3D spatial data allows efficient processing in deep learning and is used in many tasks, such as shape completion to estimate the complete geometry of objects from partial observations and semantic segmentation in real-world 3D environments. PointNet [71] was proposed to directly process point clouds effectively, respecting the permutation invariance of points in the input. A hierarchical neural network PointNet++ [72] was further introduced to solve a problem in PointNet: its inability to capture local structures induced by the metric space points live in. This enables robust recognition of local fine-grained patterns with increasing contextual scales and generalisability to complex scenes in classification and semantic segmentation tasks. Point Transformer [73] is a simple, fast, and effective model in semantic segmentation of both indoor and outdoor scenarios, overcoming the drawback of existing point cloud-based methods that require the same size of point clouds as input in the same batch for training. By harnessing the power of scale, it also gets beyond the current trade-offs between accuracy and efficiency in the context of point cloud processing. To generate fine-grained shape completions, point completion network (PCN) [74] was presented. This method works directly on raw point clouds without requiring any structural assumptions or annotations on the underlying form. SnowflakeNet [75] further improved the recovery of fine local geometric details by introducing a snowflake point deconvolution (SPD) block. The creation of point clouds is modelled by SPD as the snowflake-like development of points, with each SPD being followed by gradual generation of offspring points via separating their parent points. In SPD, the skip-transformer is introduced to learn the point splitting patterns that are the most appropriate for the local areas. Previous work processes 3D data using either voxel-based or point-based neural network models. VoxelNet [76] was proposed as a generic 3D object detection framework, that encodes a point cloud as a descriptive high-dimensional volumetric representation. The VoxelNet takes the advantages of both the sparse point structure and efficient parallel

processing based on the voxel grid. Liu *et al.* [77] proposed a point-voxel CNN model that represents the 3D input data in points to reduce the memory consumption and enable resolution scale-up, while applying the convolutions in voxels to enhance locality and decrease irregular, sparse data access. This model, with a combination of both voxel- and point cloud-based representations, demonstrated improved performance than the methods handling each of them in segmentation tasks.

2.2.2 General 3D Reconstruction from 2D Images

3D Object Reconstruction

Deep learning in 3D object reconstruction focuses on using neural networks to create detailed 3D models of objects from 2D images. 3D object reconstruction methods try to reconstruct objects from a single-view image [78–87], multi-view images [88, 89], or both [90–93]. They usually reconstruct objects of volumetric grid representation [81, 82, 88–91, 93], or some other forms such as mesh or point cloud [78–80, 83–87, 89, 92]. Most approaches rely on supervised learning [79–82, 86, 88, 89, 91, 93], while some methods attempt to solve the problem when 3D labels are not available [83–85, 87, 90, 92]. The taxonomy for deep learning-based 3D object reconstruction methods is illustrated in figure 2.9.

In [80], an end-to-end graph-based CNN architecture was presented, that uses perceptual information taken from the input picture to gradually deform an ellipsoid to create a 3D shape in a triangular mesh from a single-colour image. The coarse-to-fine strategy adopted in this method makes the whole procedure stable and guarantees physically accurate 3D geometry with better details. Mescheder *et al.* [78] presented an expressive representation for deep learning algorithms in 3D reconstruction called Occupancy Networks. Occupancy Networks implicitly represent the 3D surface as a continuous decision boundary of a classifier. The experiments illustrated effectively the 3D reconstruction performance from single images, noisy point clouds, coarse discrete voxel grids, and unconditional mesh

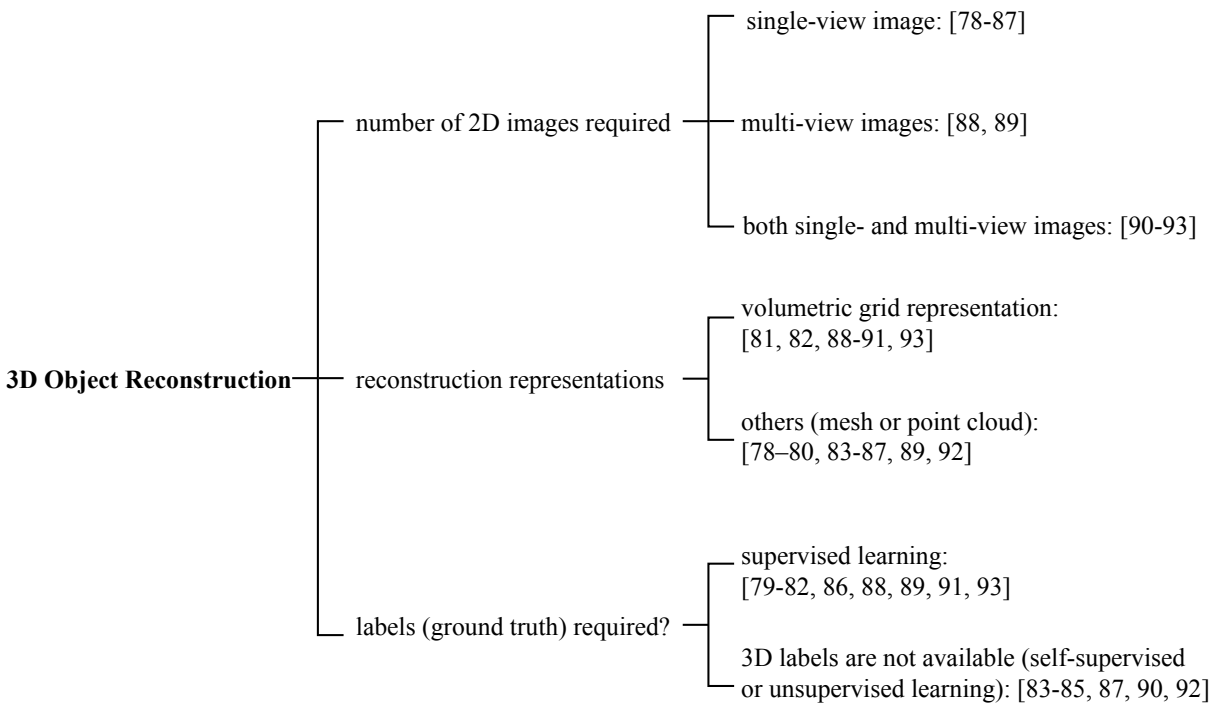


Figure 2.9: Taxonomy for deep learning-based 3D object reconstruction methods.

generation. Bian *et al.* [79] further proposed Ray-ONet to improve Occupancy Networks by a series of predictions of occupancy probabilities along a ray, which is backprojected from one 2D image pixel in the camera coordinate. This largely reduces the network inference complexity with more than 20× speed-up compared to Occupancy Networks, while maintaining a similar memory footprint during inference.

Choy *et al.* [91] proposed a 3D recurrent reconstruction neural network (3D-R2N2), which learns feature mappings from 2D objects' images to their underlying 3D structures in the representation of a 3D occupancy grid. Extensive experiments on multiple large public datasets illustrated this model's ability for reconstruction from one or multiple images of an object model based on random viewpoints. The results also showed the model could overcome the challenges of images with inadequate texture or wide baseline viewpoints. However, recurrent neural network (RNN)-based approaches are not able to provide consistent reconstruction results if the order of the same input images changes. Furthermore, RNNs may miss crucial features from early input images because of long-term memory loss. To solve these problems, Xie *et al.* [93] proposed a framework with a multi-

scale context-aware fusion module and a refiner for 3D object reconstruction from one or more views, named Pix2Vox++. The performance of the Pix2Vox++ is computationally efficient: about 7× faster than 3D-R2N2 with respect to inference time in single-view reconstruction. Yang *et al.* [88] proposed an efficient and robust model to attentively aggregate a randomly sized deep feature set for 3D reconstruction from multiple views. This model is made up of two parts. The first is a permutation-invariant feed-forward neural module that is computationally efficient and flexible to implement. The second module is a dedicated training algorithm, which is robust and able to generalise to a random number of input images. Tang *et al.* [89] used a topology-preserved, volumetric skeletal shape representation to improve object surface reconstruction of complex topologies via a bridged learning of a skeletal point set. They presented a differentiable point-to-voxel layer to enable end-to-end training. Tahir *et al.* [81] presented a method claimed to be the first to apply VAEs to 3D reconstruction from a single 2D image. In this method, the encoder learns a suitable compressed latent representation with a discriminative set of features from 2D images, and the decoder generates a corresponding smoother and high-resolution 3D model. Xing *et al.* [82] introduced a memory prior contrastive network, which can store shape prior knowledge in a few-shot learning-based single-view 3D reconstruction framework. It can handle the inter-class variability without category annotations.

Supervised 3D reconstruction has shown dramatic progress with deep neural networks, but this improvement in performance needs large-scale annotations of 2D/3D data. The first attempt for 3D representation inference from one or multiple 2D images in a purely unsupervised manner was proposed in [92]. This work exploits deep generative models of 3D structures and recovers these structures from 2D images via probabilistic inference in an end-to-end training manner without any ground truth. The unsupervised learning in the work is achieved by incorporating a 3D-to-2D neural projection layer for learning. Gwak *et al.* [90] used foreground masks as weakly supervised learning via a raytrace pooling layer, enabling perspective projection and backpropagation that links the representation gap between 2D masks and 3D volumes. This work imposes an adversarial constraint on

the reconstruction results to be matched with mask observations. The performance results on single- and multi-view reconstruction showed successful reconstruction of a high-quality object volume under 2D weak supervision with reconstruction accuracy comparable to the previous work that utilised 3D full supervision. Additionally, this study hinted at greater practical value by demonstrating a substantial generalisation capability for single-view image reconstruction with noisy viewpoint estimation. Navaneet *et al.* [83] only needed an image collection of an object category and the related silhouettes to learn 3D point cloud reconstruction from a single image in a 2D self-supervised way using a differentiable point cloud renderer. More test-time optimisation is possible with the 2D supervision, and the outcomes showed that even with less supervision, competitive performance may be attained when compared to multi-view supervised methods. Li *et al.* [85] claimed to be the first to tackle the single-view 3D reconstruction issue without the use of semantic keypoints or a category-specific template mesh. By successfully ensuring semantic coherence between the reconstructed meshes and the original photos, they were able to develop a self-supervised, single-view 3D reconstruction model that uses a collection of 2D images and silhouettes to predict the 3D mesh form, texture, and camera attribute of a target item. Hence, their model performed at least as well as category-specific reconstruction techniques that are learnt under supervision, and it can readily generalise to different object categories without labels. Hu *et al.* [84] enhanced the 3D mesh attribute learning process by proposing a self-supervised mesh reconstruction method from a single-view image, only requiring silhouette mask annotation.

Recently, there have been several methods leveraging the strong generative ability of diffusion models in the 3D reconstruction task. Melas *et al.* [87] proposed RealFusion that leverages a diffusion-based 2D image generator to reconstruct a full-view model of an object from a single image, without special 3D supervision. They engineered a prompt to the conditional diffusion-based generator pretrained on large datasets and encouraged it to 'dream up' novel views of the object. Then the full 3D object is reconstructed based on NeRF from the original image and the 'dream-up' images generated by the diffusion

model. RealFusion demonstrated better reconstruction performance compared to prior methods for monocular 3D reconstruction of objects, including the side of the object not visible in the image. Wonder3D was presented by Long *et al.* [86] to effectively create high-fidelity textured meshes from single-view photos. To enhance performance, 3D supervision guiding of normal map production during training is used. To increase the overall quality, consistency, and efficiency of single-view reconstruction tasks, Wonder3D introduced a cross-domain diffusion model that produces multi-view normal maps and the related colour pictures. In comparison to previous efforts, their assessments showed good efficiency, strong generalisation, and high-quality reconstruction outcomes.

3D Scene Reconstruction

The goal of 3D scene reconstruction is to use multiple views of a scene taken from various angles in order to restore the scene's dense 3D structure. Recently, many deep learning methods have achieved impressive performance in this task or novel scene view synthesis without depth map estimation [94]. These algorithms can be mainly categorised into voxel-based [95, 96], NeRF-based [40, 97–100], Gaussian splatting-based [101], and large feed-forward methods [102–104]. An overview of the pros and cons of these categorised methods is illustrated in figure 2.10.

Typically, voxel-based techniques use implicit functions like signed distance function (SDF) to approximate the scene geometry volumetric representation. An end-to-end 3D scene reconstruction technique was described by Murez *et al.* [95], which includes directly regressing a truncated SDF from a collection of posed photos. The first learning-based framework for real-time 3D scene reconstruction from a monocular video was proposed by Sun *et al.* [96], a neural network which can successively reconstruct local surfaces represented as sparse truncated SDF volumes for each video fragment. The high computational cost of the voxel-grid volume limits the 3D reconstruction's resolution.

NeRFs [40] showed a new emerging 3D representation by implicit neural networks. In order to oversee a 3D radiance-based representation with 2D image-level losses, NeRFs use a

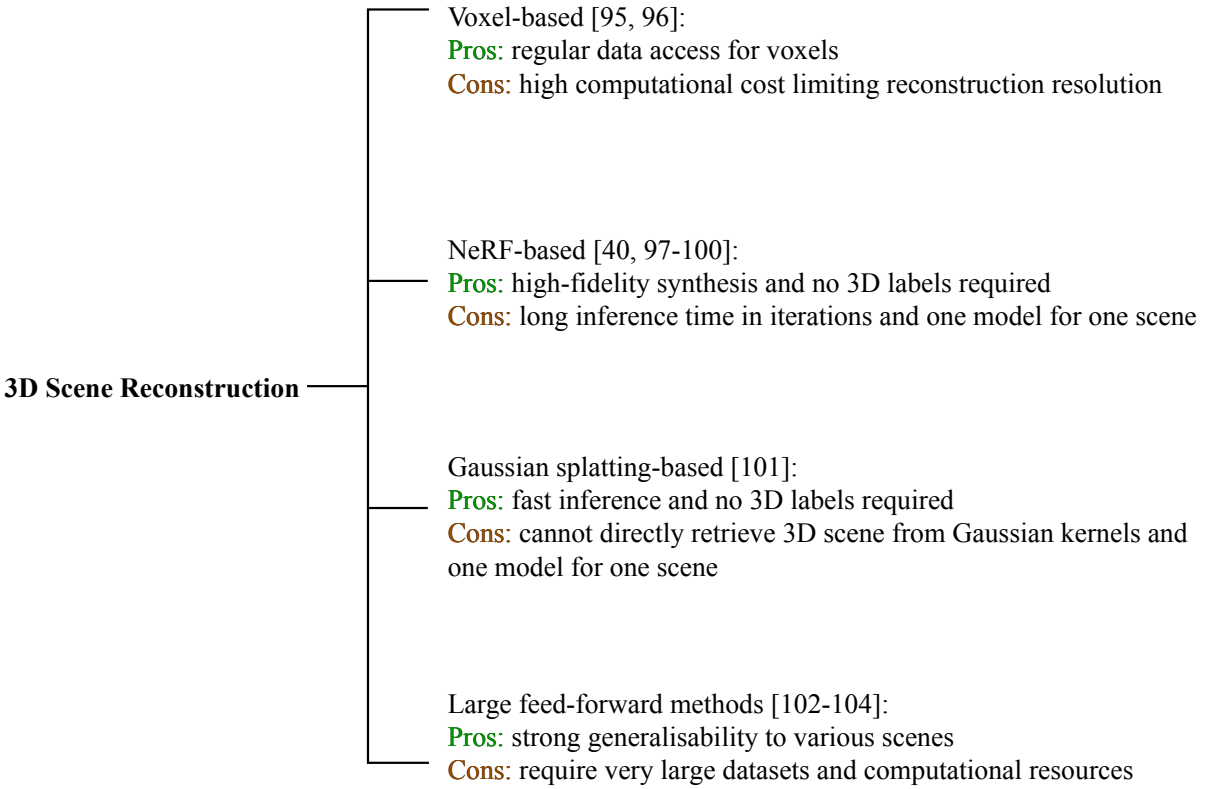


Figure 2.10: Taxonomy for deep learning-based 3D scene reconstruction methods: the pros and cons.

differentiable volume-rendering technique and an MLP to map a position and the normalised view direction to the matching colour and volume density. Novel view synthesis is NeRFs' primary goal. In [97, 98], NeRFs and SDF were combined for surface reconstruction, where the SDF is converted into densities for volume rendering, to facilitate 3D reconstruction. To speed up training and enhance surface details, Müller *et al.* [99] suggested instant neural graphics primitives with a multiresolution hash encoding using hash grids. NeRFs [40] generate high-quality view synthesis results, but are optimised per-scene, resulting in unsatisfactory reconstruction time. Xu *et al.* [100] proposed Point-NeRFs that use neural 3D point clouds with related neural characteristics to construct a radiance field. In a rendering pipeline based on ray marching, rendering can be produced effectively by aggregating neural point features close to scene surfaces. Furthermore, it can be started by directly inferring a pretrained deep network to create a neural point cloud. This point cloud can be fine-tuned to outperform NeRFs in terms of visual quality, with 30× less

training time. Through a unique pruning and growing process, it can handle the faults and outliers in existing 3D reconstruction methods when used in conjunction with them.

In contrast to implicit representation NeRFs [40] with a coordinate-based MLP, 3D Gaussian splatting (3DGS) [101] explicitly represents the scene with point primitives, each of which is parameterised as a scaled Gaussian kernel with a mean, a 3D covariance matrix, a colour modelled by Spherical Harmonics, and an opacity. Different to NeRFs using volume rendering, 3DGS uses tile-based rasterisation to efficiently render the scene. 3DGS minimises the rendering loss same as NeRFs. The initialisation of point primitives is crucial for 3DGS's performance; initialisation from structure from motion or multi-view stereo performs better than random initialisation. Although 3DGS achieved high-quality novel-view synthesis, directly retrieving good 3D geometry from Gaussian-based representations is difficult, since 3D Gaussian kernels are oval-shaped with 3D covariance that do not correspond well to the actual surface [94].

Benefiting from large-scale transformers [39], directly learning the 3D representation from large-scale 3D datasets using large feed-forward methods is becoming popular in 3D reconstruction from single or multiple images with or without posed information. DUSt3R was proposed by Wang *et al.* [102] to directly predict pixel-aligned point coordinates from uncalibrated picture pairings. A direct extension of DUSt3R, MASt3R [103] was introduced that concentrates on the particular usage of feature description and matching across multi-view pictures. From one, a few, or hundreds of views, a straightforward and effective approach called visual geometry grounded Transformer (VGGT) [104] was introduced that can immediately infer all of a scene's important 3D features, such as camera settings, point maps, depth maps, and 3D point tracks. This is a step forward in 3D computer vision.

3D Binary Rendering

In the several aforementioned methods [40, 90, 92, 97, 98, 101] as well as other works [105, 106], various differentiable rendering approaches were designed and leveraged for

optimisation based on 2D input images using deep learning. Those input images usually provide detailed information like colour, so a variety of conditions, such as illumination and light reflection, were considered for designing these rendering methods. Several binary rendering approaches [107–112] have been proposed if 2D images of only binary structures are provided for 3D geometry reconstruction. Along a ray to determine the binary pixel on 2D projection plane, Gadelha *et al.* [107] used the exponential of the sum of all the points, Liu *et al.* [108] used the product of all the points, Li *et al.* [109] used the maximum value, Liu *et al.* [110] and Yan *et al.* [111] used maximum value alike method to determine the projected silhouette with the assumption that the light cannot cross through the object. Han *et al.* [112] used a smoothing process for rendering the binary projections. Due to limited information in 2D input images with only binary structure, it usually requires more views for successful 3D geometry reconstruction.

2.2.3 3D Reconstruction in Medical Imaging

CT imaging has been widely used in clinical diagnosis, non-invasive examination, and public safety inspection. A CT scanner adopts motorised X-ray source which performs one full circular rotation each time to reconstruct a 2D image slice of the patient, and these slices can either be displayed individually or stacked together to illustrate 3D information [113]. Limited-angle CT reconstruction is based on continuous angles of views from a subset of the full-angle data, while sparse-view reconstruction relies on a few arbitrary views that do not need to be in continuous evenly-distributed angles. Both limited-angle and sparse-view CT are under-sampled and thus have great potential to reduce radiation dose and accelerate scans. However, based on insufficient under-sampled data, reconstruction by traditional methods, such as filtered backprojection (FBP), computationally expensive iterative methods, and total variation (TV) minimisation, often results in severe streaking artifacts. Recent new approaches to reconstruction utilising deep learning algorithms have demonstrated promising results, giving them potential for clinical diagnosis [114, 115].

According to different imaging geometry protocols, the methods for under-sampled CT

reconstruction can be mainly divided into three classes, namely, parallel-beam CT, fan-beam CT, and cone-beam CT. They differ in the shapes of X-ray beams [116, 117]. The parallel and fan beams cover a slice of the scanned object during each scanning circular rotation, while cone-shaped X-ray beam can cover a volume of tissue at once. Therefore, cone-beam CT has the advantages of high speed, reduced radiation dosage, and compact size design of the scanner, but it requires careful reconstruction to avoid artifacts. In this thesis, we will focus on the cone-beam geometry-based deep learning methods.

Given 3D CT data of a patient, it is expensive to additionally collect corresponding 2D X-ray projections to form paired training data with the 3D CT ground truth, as well as unethical to subject patients to extra radiation doses. Hence, usually a digitally reconstructed radiograph (DRR) is used for simulation of a 2D X-ray projection image from CT data. A radiograph, or conventional X-ray image, is a single 2D view of total X-ray absorption through the body along a given axis. For good generalisation to real X-ray images, some style-transfer methods such as CycleGAN [61, 118, 119] can be applied to eliminate the gap between synthetic DRRs and real X-ray images. There are also some works directly using real cone-beam X-ray projections for reconstruction, such as coronary artery reconstruction based on real angiographic projections [4], which do follow the cone-beam imaging geometry pattern.

Several methods have been proposed for reconstruction from a single-view X-ray image. Henzier *et al.* [120] claimed the first application of deep learning to reconstruct a 3D volume from a single 2D X-ray image with a specialised fusion operation that enables training on low-resolution examples. Different works have applied reconstruction to different organs. Shen *et al.* [121] trained residual learning with a transformer to map a single-view projection to the corresponding 3D anatomy based on upper-abdomen, lung, and head-and-neck CT scans. Shiode *et al.* [122] developed GAN-based networks for the direct estimation and construction of highly accurate 3D bone models from X-ray. Nakao *et al.* [123] proposed an image-to-graph convolutional network to learn the relationship between shape/deformation variability and deep image characteristics according to a

deformation mapping strategy. Their results showed a successful shape reconstruction from a single-view point projection image based on liver data with respiratory motion. Yu *et al.* [124] leveraged a depth adversarial adaptation module for 3D vessel reconstruction from 2D optical coherence tomography angiography images.

In addition to single-view reconstruction, reconstruction methods based on orthogonal bi-planar systems utilising the posterior-anterior and lateral views have been attempted. Ying *et al.* [125] proposed an X2CT-GAN model using the conditional generative adversarial networks. Their method incorporated a specially proposed generator network to increase data dimension from 2D X-rays to 3D CT. One notable point for the work is that it resorted to CycleGAN to learn the genuine X-ray modality from synthetic DRR data. Based on the X2CT-GAN model, Ratul *et al.* [126] further proposed a new class-conditioned network with a transformer, which better restores density information, anatomical structure, and shape in the predicted volumes than the X2CT-GAN model. Moreover, Kasten *et al.* [127] proposed an end-to-end CNN method for 3D reconstruction of knee bones based on two bi-planar X-ray DRR images. This work also used CycleGAN for style transfer. Cafaro *et al.* [128] proposed an unsupervised generative model with prior knowledge of anatomical structures to reconstruct 3D tomographic images of the head and neck from bi-planar X-rays.

The deep learning methods for under-sampled CT reconstruction can also be classified into three groups, namely, projection domain learning [129–131], image domain learning [131–136], and learning from projection to image domain [120–128, 137–149], as concluded in figure 2.11. The aforementioned methods in the previous two paragraphs all follow the same learning manner, i.e., from projection to image domain. Shen *et al.* [131] performed learning for both projection and image domains separately. They introduced a new mechanism for using the imaging system's geometry as priors to improve the 3D volumetric CT reconstruction under ultra-sparse sampling.

The existing methods for learning only based on the projection domain are less common since the final results are 3D reconstructions in the image domain. Liang *et al.* [129]

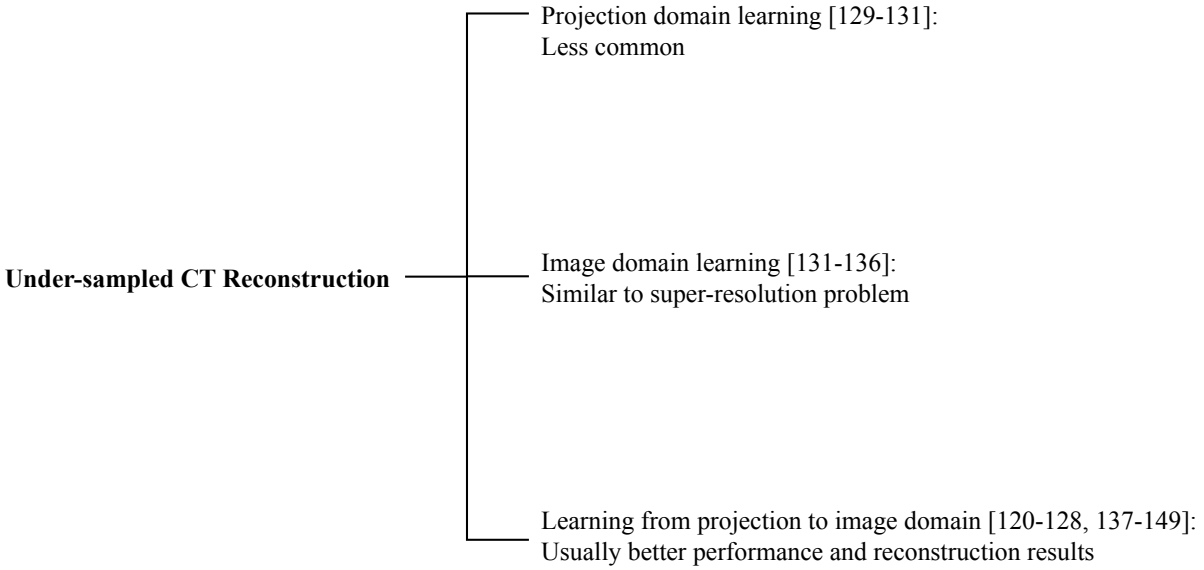


Figure 2.11: Taxonomy for deep learning-based under-sampled CT reconstruction methods.

proposed a method of angular resolution recovery in the projection domain based on deep residual CNNs to estimate the projections at unmeasured views. The reconstructed results showed little streaking artifacts and details corrupted were recovered due to the accurately recovered projections. Dong *et al.* [130] used U-Net to reconstruct CT from partial data in the projection sinogram domain. The U-Net estimated results are the full projection sinograms from a given partial projection sinogram, not the artifacts brought on by the incomplete data. Then, high-quality CT images can be reconstructed based on the complete sinogram estimations.

As for image domain learning only, the imaging geometry protocol is irrelevant and does not need to be taken into account for reconstruction. In this case, the problem will be similar to a super-resolution problem for restoring high-quality images. Jin *et al.* [132] used direct inversion followed by CNNs to solve ill-posed inverse problems that combine multiresolution decomposition and residual learning to learn to remove artifacts while keeping image structure. They demonstrated better performance of 3D reconstruction from as few as 50 X-ray views than the traditional total variation-regularised iterative reconstruction method. In [133], new multiresolution deep learning schemes via deep convolutional framelets were proposed, which are better for effective recovery of high-

frequency edges in sparse-view CT. Jiang *et al.* [134] presented a symmetric residual CNN to improve the sharpness of edges and the detail of anatomical structures based on under-sampled CT. Xie *et al.* [135] proposed a deep encoder-decoder adversarial reconstruction network for 3D spiral reconstruction. Sahu *et al.* [136] proposed an interactive framework based on CNNs for regularisation parameter tuning to solve the time-consuming problem caused by parameter-estimation dependent medical image reconstruction algorithms such as Penalised Weighted Least Squares.

Liang *et al.* [150] explored the performance differences between these three different domain learning classes. The experiments demonstrated that the three classes all performed well for sparse view CT reconstruction, while a comprehensive network combining deep learning in both the projection and image domains can achieve the best performance. However, it should be noted that the comparison here is, by some means, unfair because the comprehensive network is bigger and includes more computations. The authors [150] proposed that the end-to-end training of a comprehensive network is made possible through integrating an analytical FBP operator as one layer of the network. Würfl *et al.* [138] mapped a conventional filtered backprojection method to a neural network with a proposed differentiable cone-beam backprojection layer, effectively computing the forward pass whose backwards pass can then be derived as a projection operation. This framework can learn the discretised version of the analytically determined ramp-filter for reconstruction in an end-to-end fashion, and can propagate the gradient across the network, from the image domain to the projection domain. Based on [138], Wang *et al.* [141] additionally used a U-Net in the image domain to improve the reconstruction quality. Lagerwerf *et al.* [139] introduced a neural-network-based Feldkamp-Davis-Kress (FDK) algorithm that learns a set of FDK filters to enhance its reconstruction accuracy while retaining its computational efficiency. He *et al.* [140] proposed a unified framework for Radon inversion via deep learning from projection to image domain, where the Radon inversion as a backprojection layer can be approximated in an end-to-end fashion. Wang *et al.* [143] introduced a sinogram extrapolation module for limited-angle CT reconstruction, which

extra complements missing sinogram information and boosts model generalisability. It uses a differentiable Radon inversion layer to recreate the extrapolated sinograms after inpainting them. A multi-scale deep neural network model was used to directly learn an interpolation strategy for predicting 2D Fourier coefficients in Cartesian coordinates from the available ones in polar coordinates [142]. Neither projection nor backprojection operation is used in the method, and it demonstrated both promising performance and efficient computation.

The recent development of implicit neural representation learning has shown great improvement in medical image analysis. Neural attenuation fields with multiresolution hash encoding [144] were proposed to tackle the problem of sparse-view cone-beam CT reconstruction and demonstrated promising results in terms of both speed and quality. They can reconstruct 3D CT from as few as 50 projections without 3D ground truth. This solves the problems of many previous methods, as the acquisition of paired data has always been a concern in real clinics, but for each new CT scan, there is a need to re-optimise the model. Shen *et al.* [145] introduced prior embedding to implicit neural representation learning for reconstruction from sparsely sampled measurements. Without the need for extensive data, it creates a representation of the unknown subject by taking advantage of the internal information in an image prior and the physics of the sparsely sampled measurements. The findings demonstrated that the subtle yet important image alterations needed to evaluate tumour growth can be robustly captured by this method. Park *et al.* [146] proposed a framework for 3D teeth reconstruction from one panoramic radiograph using neural implicit functions, where the teeth are first segmented in the radiograph, followed by 3D teeth reconstruction, and 3D ground truth is involved in training. With the advantages of a large diffusion model in generating high-quality images based on a condition, a 3D teeth reconstruction framework called TeethDreamer was proposed by Xu *et al.* [147] to restore the position and shape of the upper and lower teeth based on five intra-oral photos. In order to handle sparse inputs, they used the previous knowledge of a large diffusion model to create new multi-view pictures with known postures based on the

provided shots. They then used neural surface reconstruction to create high-quality 3D tooth models. The extensive experiments demonstrated the superiority of TeethDreamer over previous methods, giving the potential to monitor orthodontic treatment remotely. In addition, a 3DGS-based method [148] was proposed, demonstrating fast inference speed, but it can only render novel view X-ray projections. This makes it inapplicable in the clinic. Zha *et al.* [149] proposed the first 3DGS-based framework for sparse-view tomographic reconstruction with accurate volume retrieval by developing a CUDA-based differentiable voxeliser.

2.3 3D Vessels Reconstruction

2.3.1 Coronary Artery Tree Reconstruction

Search Methodology

In this section, I present a comprehensive literature review for which I used four approaches: i) direct search in Google Scholar to cover a large variety of sources; the terms I used to search are "3D coronary artery tree reconstruction with/without deep learning", "limited-angle/sparse CT/X-ray reconstruction with/without vessels/artery", and "reconstruction from angiographic projections"; ii) a more exhaustive search through the databases of the MICCAI conference and journals Medical Image Analysis and IEEE Transactions on Medical Imaging to ensure no substantive contributions were missed; iii) iterative literature search from the relevant references in the papers found with the above methods; and iv) direct search for relevant review or survey papers to get a comprehensive insight in the field.

Traditional Algorithms

A substantial amount of effort has been devoted to 3D coronary artery tree reconstruction from ICA by conventional (non-machine learning) mathematical methods [4]. Different

strategies to tackle cardiac and respiratory motions have resulted in diverse coronary artery tree reconstruction methods. These conventional methods can be mainly grouped into two classes: symbolic or model-based reconstruction [20] and tomographic reconstruction [151], as illustrated in figure 2.12. The major distinction between the two classes is the reconstruction output. While model-based methods generate a 3D binary representation of coronary artery tree that includes a centreline and sometimes the vessel surface, tomographic reconstruction methods generate a 3D volume representing the coronary arteries by producing information about X-ray attenuation coefficients.

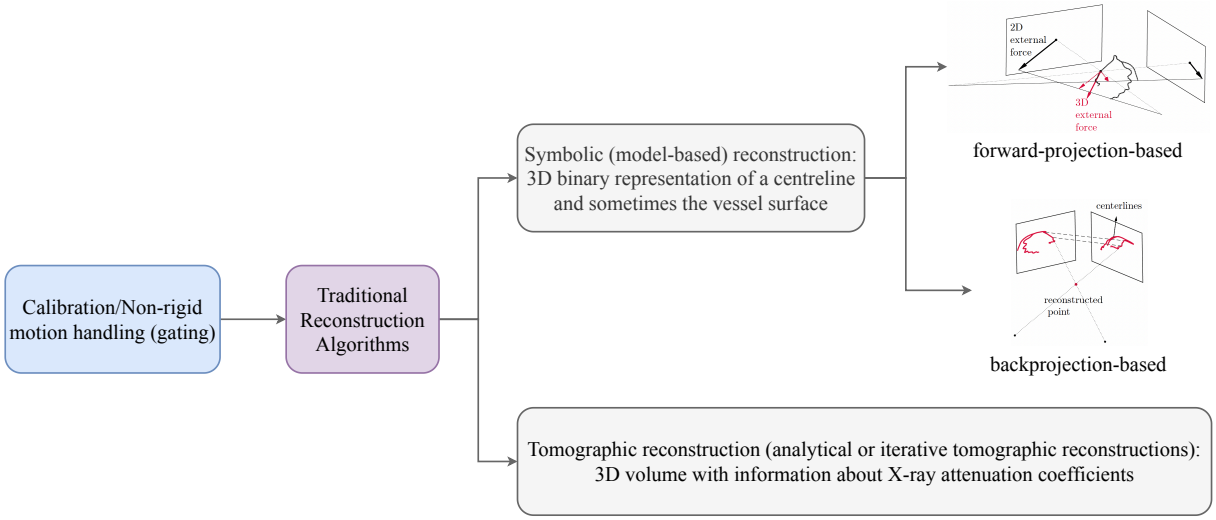


Figure 2.12: Traditional algorithms for 3D coronary artery tree reconstruction. (forward-projection and backprojection-based images taken from [4])

Non-rigid Motion Handling Despite the differences among reconstruction methods, there are several common aspects which apply to both classes. For example, the handling of challenging respiratory and cardiac motions in the coronary arteries occurring during image acquisition. Respiratory motion could be alleviated by instructing the patients to hold their breath during acquisition. Assuming no residual respiratory motion remains, retrospective gating approaches, where the image frames at the same cardiac phase are chosen, typically using the ECG, and surrogate-based gating are generally exploited to overcome the cardiac motion [4]. In the most recent reconstruction methods, the image frames are picked at the end of the diastolic phase when the heart motion is minimal,

such that it can assume no cardiac motion occurs between acquired images [152–154]. Moreover, the patient- or imaging device-related movements can be minimised using a careful protocol during acquisition. However, some residual misregistration artifacts can remain, impacting geometry conditions [155].

Model-based Reconstruction According to the design, model-based methods can be categorised into two classes: forward-projection-based and backprojection-based methods. Forward-projection-based methods for coronary artery tree reconstruction adopt a 3D model that accommodates itself to the vessel structures in 2D projections. Back-projection-based methods build the coronary artery tree from backprojection of the information extracted from 2D projection images, which are selected via ECG gating [4]. For reconstruction from limited views, non-uniform rational basis splines (NURBS) based [156, 157] and point cloud based [154] methods have been attempted. Banerjee *et al.* [154] proposed a point cloud-based method for optimally reconstructing a 3D vessel skeleton from two non-simultaneous angiographic projections by iteratively minimising the reconstruction error. This work does not require any specific image acquisition protocol. A NURBS-based method is proposed [157] for meshing complex coronary artery trees from two uncalibrated X-ray angiographic projections. In this work, an image formation model was implemented and integrated with a robust genetic algorithm optimiser to decide the calibration parameters throughout angiographic views. The frame correspondences between angiographic acquisitions were obtained using a partial-matching method.

Some of the model-based reconstruction methods [158–160] considered the possible cardiac and/or respiratory motion, while some did not [161, 162]. Merle *et al.* [161] developed a 3D reconstruction method based on a deformable skeleton model from two views to find the optimal angles of views, but without considering any motion. A method based on a bi-plane system was presented in [162], also without considering the motion effect. This method utilised the geometric properties of the X-ray acquisition system and tracking of leading edges of injected angiographic agent into each vessel for determining the corresponding

points on two projection images acquired from orthogonal views. Unberath *et al.* [158] investigated methods to assess the respiratory motion in coronary artery tree based on rotational angiograms by optimising conditions of epipolar consistency and a task-based auto-focus evaluation. Liao *et al.* [159] formulated the reconstruction problem as an energy minimisation problem including a soft epipolar line constraint and a smoothness term measured in 3D, which is robust to errors in 2D centreline extraction. Li *et al.* [160] presented 3D coronary artery tree reconstruction from two arbitrary views, using five steps of vessel segmentation, feature-points identification, vessel skeleton matching, vessel diameter quantification, and 3D reconstruction.

Tomographic Reconstruction Tomographic reconstruction methods can tackle uncommon anatomies such as collaterals and tortuous branches, due to their requirement of less or no prior information about the coronary artery trees [163]. Owing to the same reason, more accurate surface details of vessels can be provided by these methods [164]. Moreover, tomographic reconstruction approaches do not need any manual interaction, such as manual annotations. In terms of the types of tomographic reconstruction approaches, the methods can be classified as analytical and iterative tomographic reconstructions [4]. Analytical reconstruction methods take account of a simplified system model and image (volume) model. Hence, they are more appropriate for the cases where approximate solutions are sufficient. They are also well-established and fast in contrast to iterative alternatives. The FDK algorithm [116] is one of the most widely-used methods for analytical reconstruction of cone-beam geometries. Iterative reconstruction methods [165] can combine a range of acquisition geometries (*e.g.* scans in limited angles), image model, forward model, noise model, and prior information into the reconstruction [166]. Based on whether they consider a noise model or not, iterative reconstruction algorithms can be further categorised into two groups, namely algebraic and statistical methods. Both groups of methods have been used in the context of coronary artery tree reconstruction [4]. Based on how they deal with the cardiac motion, the tomographic reconstruction methods can also be divided into three groups, namely gated, motion compensated, and gated and motion compensated

methods [4]. Bousse *et al.* [167] applied a three-stage approach to deal with motion from a rotational X-ray projection sequence: i) reconstruction of the 3D coronary artery tree at different phases of the cardiac cycle; ii) motion estimation; and iii) motion-compensated tomographic reconstruction at one given phase using all available projections.

Deep Learning Approaches

Existing deep learning based-methods for 3D coronary artery tree reconstruction can be categorised into five classes according to their reconstruction representations: voxel-based [168], tubular shape-based with centreline and radii [169–171], mesh-based [172], implicit neural-based [173–177], and Gaussian-based representations [178], as summarised in figure 2.13. These methods have typically used synthetic data, CCTA data, ICA data gated at the same cardiac phase assuming no residual temporal motion, or ICA data from bi-planar scans, none of which suffer from non-rigid cardiac motion or respiratory motion between projections. This limitation makes these methods ill-suited for 3D coronary artery tree reconstruction from real non-simultaneous ICA acquisitions. In addition, there are some other works for 3D coronary artery reconstruction, in which deep learning technique is utilised only at the pre-reconstruction stage, such as vessel segmentation [179, 180] and identification of vessels' two ends [181].

Voxel Grid Representation Wang *et al.* [168] developed a weakly supervised 3D reconstruction algorithm from two angiographic views, based on WGAN. This work considers a 3D fully supervised learning framework and a 2D weakly supervised learning scheme over an original dataset of 44 3D coronary artery tree models reconstructed from CCTA images. In the 3D full supervision experiment, the dataset was augmented to $8,800$ samples for learning, and their 2D projections were regarded as input. Except for minor errors in several voxels, the predicted results were similar to the ground truth; however, artifacts and deformation were present in some reconstruction results, along with missing anatomical features. For the 2D weakly supervised reconstruction, this work

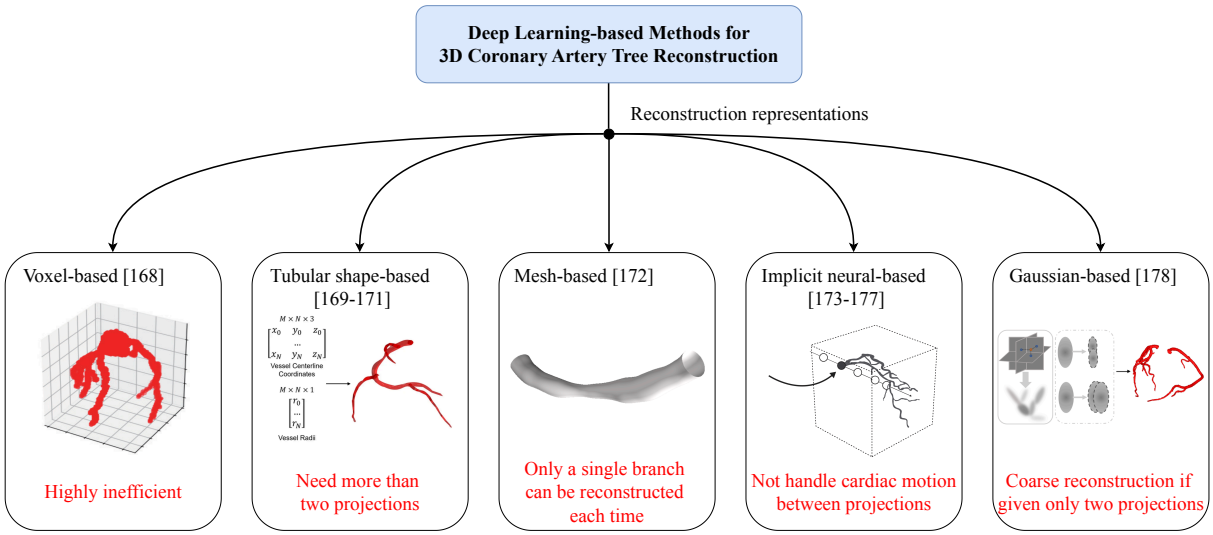


Figure 2.13: Deep learning-based methods for 3D coronary artery tree reconstruction, categorised by five reconstruction representation classes. (voxel and mesh-based images taken from [168] and [172], respectively) (tubular shape, implicit neural, and Gaussian-based images adapted from [171], [174], and [178], respectively)

divided the original 3D complete coronary artery tree models into both LAD data and RCA data in accordance with the real surgery situations, and the same augmentation as before was applied to generate $8,800$ samples for each of them. The authors inserted a ray-trace pooling layer [90] into the generator network to realise a reprojection layer for rendering two 2D images from the 3D reconstructed results based on the angiographic imaging procedure and given camera parameters. Thus, weakly supervised learning can then be accomplished by both the 3D supervised reconstruction and calculation of the cross-entropy loss between rendered 2D images and input images. This work does not take into account any motion between projections, and the voxel-based representation using deep learning is highly inefficient due to the sparse structure of vessels in 3D volume.

Tubular Shape-based Representation Atli and Gedik [169] explored the potential use of deep learning in the 3D coronary artery tree reconstruction problem. They randomly created 10,000 synthetic 3D samples based on real coronaries of 9 subjects for training, thus providing 3D coronary artery tree ground truth for evaluation. They also proposed a new efficient data structure for the representation of coronary artery tree in a tubular shape, instead of a voxel-grid shape. Three models were adopted for evaluation, namely, multi-

view fully CNN, time distributed auto-encoder CNN, and auto-encoder followed by long short-term memory (LSTM). The inputs to these models are multi-view segmented X-ray images, and the output is structured tubular representation. Their results demonstrated a promising leverage of deep learning architectures for 3D coronary artery tree reconstruction. Similar to this, in [170], a 3D synthetic coronary artery tree was reconstructed using synthetic segmented 2D X-ray angiography images, based on a fully connected CNN model. A multi-stage neural network method for reconstructing 3D RCA coronary artery trees from uncalibrated 2D ICA images without any knowledge of image acquisition parameters was presented by Iyer *et al.* [171]. The approach comprises distinct phases for reconstructing the vessel's radius and centreline, as well as a single backbone network. An analytical matrix representation of the coronary tree as an output is suitable for downstream applications. This network was also trained based on a synthetic dataset. The aforementioned three works only incorporate synthetic data, which do not contain real patients' anatomies and do not consider cardiac or respiratory motion that could heavily impact the final performance. Also, their reconstruction is based on more than two projections.

Mesh-based Representation Bransby *et al.* [172] proposed 3DAngioNet, a graph convolutional network-based approach that enables rapid 3D vessel mesh reconstruction from bi-planar ICA views. However, this work can only reconstruct a single coronary segment each time, and there is no cardiac motion between projections.

Implicit Neural-based Representation The possibilities and limitations of using NeRFs for 3D reconstruction from X-ray angiography were examined by Maas *et al.* [173]. A thorough experimental analysis shows that NeRFs have the potential for 3D X-ray angiography reconstruction from sparse or limited-angle projections, but it also identifies certain issues that need to be addressed beforehand, such as the overlap of background structures. Maas *et al.* [174] expanded on this work by introducing NeRF-CA, the initial step toward 4D coronary artery reconstruction, which accomplishes reconstructions from

sparse coronary angiograms with cardiac motion. Through a novel combination of scene decomposition and regularisation restrictions that suit the sparse character of the blood vessel structure, they isolated the coronary angiography scene in a dynamic coronary artery from a static background. These methods impose dynamic structural sparsity and picture smoothness, which separate the coronary artery from background. 4D reconstructions from as few as 4 angiography sequences are made possible by a special combination of these techniques. However, only two 3D data were used for testing and projection simulation. Moreover, although it is for 4D reconstruction, they do not consider cardiac motion between projections. Instead, they simulate at least four projections at the same cardiac phase with the assumption of no residual temporal motion to do 3D reconstruction and exhibit cardiac motion by 3D reconstruction from different phases. Built on top of NeRF-CA [174], NerT-CA [175] was further proposed. With sparse-view coronary angiography, this hybrid method uses neural and tensorial representations to speed up 4D reconstructions. Using neural fields for dynamic sparse reconstruction and fast tensorial fields for low-rank static reconstruction, the coronary angiography image is modelled as a decomposition of low-rank and sparse components. This approach outperformed previous work [174] in both reconstruction time and accuracy, allowing for efficiently reconstructing dynamic coronary artery from as few as three coronary angiogram views. However, cardiac motion between projections is still not considered. In [176], a NeRFs-based architecture was used to generate novel views of coronary arteries, but only static data can be handled, so dynamic artifacts caused by pulsing vasculature are unsolved. Moreover, a further step using traditional reconstruction methods is required to obtain the final 3D reconstruction given novel view synthesis. AutoCAR, a completely automated transfer learning-based system for sparse 3D dynamic cardiovascular reconstruction, was presented by Zhu *et al.* [177]. Sparse backwards projection, vascular graph optimisation, and pose domain adaptation are the three primary parts of AutoCAR. Using the inherent spatial sparsity of cardiac vessels for computational design and combining the X-ray angiography imaging parameter statistics of more than $1,000$ clinical cases into synthetic data generation, AutoCAR

allows dynamic cardiovascular reconstruction in actual clinical settings. The dynamic reconstruction by AutoCAR is similar to [174, 175], so it has the same issue of not being able to handle cardiac motion between projections.

Gaussian-based Representation 3DGR-CAR is the first effort to employ Gaussian representation for coronary artery reconstruction from ultra-sparse 2D X-ray projections, as proposed by Fu *et al.* [178]. They suggested a Gaussian centre predictor based on U-Net to get over the noisy Gaussian initialisation from ultra-sparse view projections and utilised 3D Gaussian representation to overcome the inefficiency brought on by the extreme sparsity of coronary artery data. However, it needs more than half an hour for reconstruction, and the reconstruction results from two views are less satisfactory. This work also does not take real clinical settings into account and no motion between projections is considered.

Commercial Software

There exist several commercial software packages for 3D coronary artery tree reconstruction. For example, Siemens Medical Solutions released a product called Interventional Cardiac 3D software (IC3D) [182] in 2004, which can create a volumetric reconstruction quickly and efficiently without the requirement of rotational angiography. However, it is based on two single static images by a bi-planar imaging system, and so, cardiac motion is not considered. Philips released a similar product called Allura 3D Coronary Angiography (3D-CA) [183] in 2004 as well, that can generate 3D images of diseased coronary vessels in a few seconds from multiple viewpoints and angles achieved during a single rotational angiography run. It may help to prevent misrepresentations of lesions and bifurcations by minimising foreshortened views of the coronary vessel tree. Apart from the limitation of requiring a rotational angiography, the Philips product has no mention of the handling of cardiac motions, and its underlying methodology is not publicly available. Another product called Medis Suite XA [184] is used for 3D quantitative coronary angiography

reconstruction, which provides the anatomical parameters for stent sizing and optimal viewing angles for visualising the lesion area. It is suitable for online use during patient procedures and can be applied to monoplane and bi-plane acquisitions. However, it uses an epipolar geometry-based 3D reconstruction approach, which is prone to errors such as foreshortening and false reconstructions in curved vessels. HeartFlow [185] is another commercial software that uses CT scans to create a 3D image of the heart and uses artificial intelligence to predict the impact of any blockages of the arteries. This technology, based on CT scans, shows the potential to reduce unnecessary and invasive diagnostic coronary angiography procedures, which could cause bleeding and major vessel injury. However, compared to ICA-based methods, this technology would cause higher radiation dosage due to a fully circular scan and provide less accurate resolution of the luminal surface.

In addition, there are two commercially available CT image reconstruction methods leveraging deep learning models cleared by the FDA: TrueFidelity [186] by GE Healthcare and AiCE [187] by Canon Medical Systems [188]. TrueFidelity demonstrated the potential to reduce dosage by up to 56% while achieving similar detectability with a hybrid iterative reconstruction and deep learning strategy. When evaluating the detectability of hypovascular hepatic metastasis in reconstructed results, AiCE showed reduced image noise and superior conspicuity compared to iterative reconstruction. Both TrueFidelity and AiCE can produce clean, high-fidelity images from noisy, low-dose raw data, because they use a neural network trained on a vast dataset of high-quality images to identify and suppress noise patterns while preserving and enhancing true anatomical structures [188]. Hence, they allow radiologists to significantly lower the radiation settings (such as tube current) on the CT scanner for routine CT examinations to achieve safer medical imaging. This could result in a practice change that a protocol which previously required a certain dose to achieve acceptable image quality can now be run at a much lower dose. Although radiologists highly rated TrueFidelity and AiCE for subjective image quality, this must be regarded with caution. First, further external validation of deep learning image reconstruction is required. Second, while the potential for dose reduction has been

demonstrated in phantoms and patients, real dose reduction in the clinical routine as a result of changing acquisition settings while performing diagnostic investigations has yet to be validated. This is to say, in controlled experiments using phantoms and specific research studies involving patients, they have successfully shown that they can reconstruct clear images from low-dose data, but we have not yet proven that hospitals are actually changing their protocols to turn down the radiation settings in their daily workflow [188]. A human operator (or an automated protocol) must actively go into the CT console and lower the tube current (mA) or voltage (kV) to make real dose reduction happen, so without a deliberate, validated change in hospital policy and protocol, the mere presence of the software does not guarantee the patient receives less radiation. Instead, clinicians might prefer the new, superior image quality and keep the radiation settings unchanged, so the software may be used to improve image quality rather than to reduce radiation [188]. Third, the decision-making process of such deep learning algorithms is opaque to human perception. Moreover, these two softwares are designed specifically for CT reconstruction, so they cannot be used for our problem of coronary artery tree reconstruction.

2.3.2 Cerebral Vessels Reconstruction

In addition to 3D coronary artery tree reconstruction, the reconstruction of cerebral vessels has been attempted in several works. The methods for 3D cerebral vessels reconstruction can be mainly classified into two classes: centreline reconstruction with either radius or cross-sectional area and lumen reconstruction in voxel- or surface-based representation [189]. Most algorithms to reconstruct the vessel centreline and obtain its radius were initially proposed for quantitative coronary angiography, but they could also be applied for the reconstruction of cerebral vessels. A point-based reconstruction method [190] using bi-planar projections was explored for cerebral vessels reconstruction, where the corresponding points in two X-ray views from different orientations or locations must be determined. Point-based reconstruction methods cannot accurately define the depth of vessels based on two projections and need manual annotations for the corresponding points if geometry of

the X-ray system is unknown. There exist some works that reconstruct the 3D voxels of cerebral vessels from X-ray projections [191–194]. Niki *et al.* [191] proposed an effective 3D reconstruction method of cerebral blood vessels based on an X-ray rotational angiographic system using the cone-beam filtered backprojection algorithm, where the imaging system acquires 60 angiograms over 180 degrees. Iterative methods have been exploited for 3D reconstruction [192, 193]; but they are time-consuming and hence, not applicable for routine clinical use. Schueler *et al.* [194] described an algebraic reconstruction technique to produce an enhanced 3D reconstruction of cerebral vasculature from X-ray projections. However, all these methods [192–194] are based on 6 projections acquired in bi-plane C-arm systems, while our objective for 3D coronary artery tree reconstruction is based only on two non-simultaneous projections acquired in single-plane systems.

Apart from the methods based on mathematical models reconstructing the 3D cerebral vessels, the use of deep learning models has been prevalent to tackle this problem in recent years. An adversarial network for 3D neurovascular reconstruction based on bi-plane angiograms was built by Zuo [195]; however, the results are not satisfactory, with noticeable defects appearing among the crossed arteries. A self-supervised learning model [196] was presented for the 3D cerebral vessel reconstruction from ultra-sparse projections. This method demonstrated high diagnostic accuracy and much less radiation dosage exposure compared to the gold standard method of FDK-based algorithms, while still requiring 8 projections. Cafaro *et al.* [197] used bi-planar DRRs for cerebral vascular reconstruction. They first generate a 3D coarse backprojection from two projections, which is then sent to a U-Net to obtain an initial reconstruction result. A maximum a posteriori approach with a continuity-favouring prior is later used to iteratively refine this reconstruction result, yielding a high degree of reconstruction accuracy and indicating that as few as two projections could be enough to disambiguate structures for precise 3D reconstruction. The first Gaussian splatting-based framework, 4D radiative Gaussian splatting (4DRGS), was introduced by Liu *et al.* [198] for effective 3D vessel reconstruction from sparse-view dynamic cerebral digital subtraction angiography (DSA) images. In order to describe

static vessel structures, the vessels are represented as 4D radiative Gaussian kernels with time-invariant geometry parameters, such as position, rotation, and scale. To capture the temporally variable response of contrast agent flow, a compact neural network predicts the time-dependent central attenuation of each kernel. Through X-ray rasterisation, these Gaussian kernels are splattered to create DSA images, which are further optimised using actual captured ones. The well-trained kernels are used to voxelise the final 3D vascular volume. To further enhance reconstruction quality, bounded scaling activation and accumulated attenuation pruning are included. The 4DRGS demonstrates impressive state-of-the-art (SOTA) results and much faster reconstruction speed. However, it has limitations: the assumption of no patient movement during scanning, no consideration of calibration errors in scanner geometry and imaging noise in the scanning process and multi-view scans required (at least 30 views). In addition to reconstruction from X-ray, CT reconstruction using deep learning has also been explored [199, 200], showing lower noise and improved tissue differentiation compared to filtered-backprojection and hybrid-iterative reconstruction. However, these CT reconstruction methods are based on slices instead of X-ray projections.

Compared to the coronary arteries, the cerebral vessels are mostly stationary, so the reconstruction methods for the cerebral vessels cannot be directly applied to the coronary arteries, as they would fail due to unmodelled cardiac motion.

2.4 Evaluation

2.4.1 Metrics

We can directly measure the 3D reconstruction results if we have corresponding 3D ground truth, such as when using CCTA data. In real clinical scenarios, when there is no corresponding 3D ground truth to real ICA data, we can measure the reprojections of 3D reconstruction results with the 2D ICA data. In some studies [154], for a robust clinical evaluation, optical coherence tomography (OCT) data are obtained where the 3D coronary

artery tree reconstruction from ICA data is evaluated against the OCT data collected on the same patient. The OCT evaluation allows accurate vessel lumen comparison.

Let us assume $\widehat{\mathbf{y}}$ as the 3D coronary artery tree reconstruction results, $\mathbf{y}$ as the corresponding binary ground truth, and $bin_{\varsigma}(\widehat{\mathbf{y}})$ as the binarised transformation of $\widehat{\mathbf{y}}$ with a threshold of ς.

Dice Similarity Coefficient (*Dice*) Dice similarity coefficient (*Dice*), also known as the Dice score, is a common evaluation metric in the field of medical image analysis to assess the similarity or overlap between two sets, which are usually represented as binary data. It measures how well the predicted area aligns with the ground truth area. It is calculated based on the double number of intersected elements divided by the total number of elements between the predicted result and ground truth, as:

$$Dice(\widehat{\mathbf{y}}, \mathbf{y}) = 2 \times \frac{|bin_{\varsigma}(\widehat{\mathbf{y}}) \cap \mathbf{y}|}{|bin_{\varsigma}(\widehat{\mathbf{y}})| + |\mathbf{y}|}. \tag{2.1}$$

Dice ranges from 0 to 1, where a bigger value suggests a better performance on how well the model captures spatial extent and boundaries of the coronary tree. $Dice = 0$ indicates no overlap or similarity between the two sets, meaning the predicted result has no common elements with the ground truth, while $Dice = 1$ represents perfect overlap, where the predicted result is identical to the ground truth.

Intersection Over Union (*IoU*) Intersection over union (IoU), also known as the Jaccard index, is the most commonly used metric in the field of computer vision, such as object detection, to describe the extent of overlap between two arbitrary shapes. It is calculated based on the ratio of intersection between predicted result and ground truth over their union as:

$$IoU(\widehat{\mathbf{y}}, \mathbf{y}) = \frac{|bin_{\varsigma}(\widehat{\mathbf{y}}) \cap \mathbf{y}|}{|bin_{\varsigma}(\widehat{\mathbf{y}}) \cup \mathbf{y}|}. \tag{2.2}$$

IoU can have any values between 0 and 1, where the greater the region of overlap, the greater the IoU. When there is no intersection between the predicted result and ground truth, the IoU equals 0, while IoU equals 1 when they are completely overlapped.

Dice and *IoU* are monotonically related, with *IoU* values being smaller than *Dice*:

$$IoU = \frac{Dice}{2 - Dice}, Dice = \frac{2 \times IoU}{1 + IoU}. \tag{2.3}$$

The schematic illustrations of the computation of both *Dice* and *IoU* metrics are shown in figure 2.14.

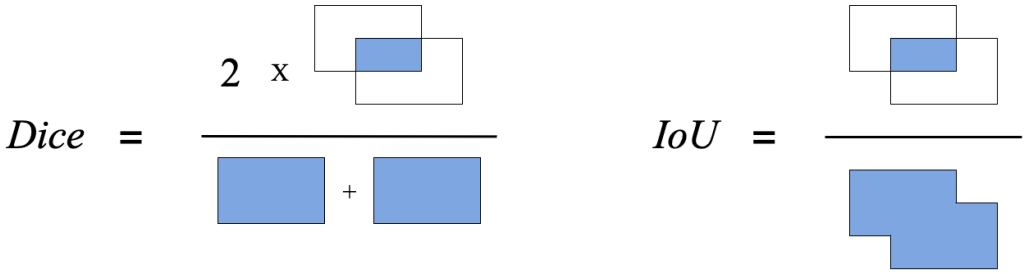


Figure 2.14: Schematic illustrations of the computation of both *Dice* (left) and *IoU* (right) metrics.

Centreline Dice Score (*clDice*) The coronary artery has a specific shape, and its anatomical structure is important during evaluation. Centreline Dice score (*clDice*) metric [201] measures the similarity between vessels-alike objects based on their topology characteristic, and is particularly useful for checking connectedness preservation. It is calculated based on the intersection of object masks and their (morphological) skeleton, and has been proven for topology preservation. First, the skeletons $\mathcal{S}_{bin_\varsigma(\widehat{\mathbf{y}})}$ and $\mathcal{S}_\mathbf{y}$ are extracted from $bin_\varsigma(\widehat{\mathbf{y}})$ and $\mathbf{y}$ respectively. Subsequently, we compute the fraction of $\mathcal{S}_{bin_\varsigma(\widehat{\mathbf{y}})}$ that lies within $\mathbf{y}$, which we call *Topology Precision* or $\mathcal{T}prec(\mathcal{S}_{bin_\varsigma(\widehat{\mathbf{y}})}, \mathbf{y})$, and vice-a-versa we obtain *Topology Sensitivity* or $\mathcal{T}sens(\mathcal{S}_\mathbf{y}, bin_\varsigma(\widehat{\mathbf{y}}))$. The calculation is defined as:

$$\begin{aligned} \mathcal{T}prec(\mathcal{S}_{bin_\varsigma(\widehat{\mathbf{y}})}, \mathbf{y}) &= \frac{|\mathcal{S}_{bin_\varsigma(\widehat{\mathbf{y}})} \cap \mathbf{y}|}{|\mathcal{S}_{bin_\varsigma(\widehat{\mathbf{y}})}|}, \\ \mathcal{T}sens(\mathcal{S}_\mathbf{y}, bin_\varsigma(\widehat{\mathbf{y}})) &= \frac{|\mathcal{S}_\mathbf{y} \cap bin_\varsigma(\widehat{\mathbf{y}})|}{|\mathcal{S}_\mathbf{y}|}. \end{aligned} \tag{2.4}$$

The measure $\mathcal{T}prec(\mathcal{S}_{bin_\varsigma(\widehat{\mathbf{y}})}, \mathbf{y})$ is susceptible to false positives in the prediction, while the measure $\mathcal{T}sens(\mathcal{S}_\mathbf{y}, bin_\varsigma(\widehat{\mathbf{y}}))$ is susceptible to false negatives. This explains the rationale behind referring to the $\mathcal{T}prec(\mathcal{S}_{bin_\varsigma(\widehat{\mathbf{y}})}, \mathbf{y})$ as topology's precision and to the $\mathcal{T}sens(\mathcal{S}_\mathbf{y}, bin_\varsigma(\widehat{\mathbf{y}}))$ as its sensitivity. Since we want to maximize both precision and sensitivity (recall),

clDice is then defined to be the harmonic mean of both measures $\mathcal{T}prec(\mathcal{S}_{bin_\varsigma(\widehat{\mathbf{y}})}, \mathbf{y})$ and $\mathcal{T}sens(\mathcal{S}_{\mathbf{y}}, bin_\varsigma(\widehat{\mathbf{y}}))$, as:

$$clDice(\widehat{\mathbf{y}}, \mathbf{y}) = 2 \times \frac{\mathcal{T}prec(\mathcal{S}_{bin_\varsigma(\widehat{\mathbf{y}})}, \mathbf{y}) \times \mathcal{T}sens(\mathcal{S}_{\mathbf{y}}, bin_\varsigma(\widehat{\mathbf{y}}))}{\mathcal{T}prec(\mathcal{S}_{bin_\varsigma(\widehat{\mathbf{y}})}, \mathbf{y}) + \mathcal{T}sens(\mathcal{S}_{\mathbf{y}}, bin_\varsigma(\widehat{\mathbf{y}}))}. \tag{2.5}$$

Note that the *clDice* formulation is not defined for $\mathcal{T}prec = 0$ and $\mathcal{T}sens = 0$, but can easily be extended continuously with the value 0. $clDice(\widehat{\mathbf{y}}, \mathbf{y}) \in (0, 1]$ where a bigger value suggests a better performance in vessel topology preservation.

Overlap using A Sweeping Distance Threshold ($Ot(d)$) Overlap using a sweeping distance threshold ($Ot(d)$) is an application-specific evaluation metric, where d is the distance threshold in mm unit, as proposed in 'A Coronary Artery Reconstruction Challenge' [202], which can tackle the deformation in 3D reconstructions, unavailability of 3D ground truth for real ICA scans, and motion between projection planes. $Ot(d) \in [0, 1]$ with 0 representing no overlap and 1 the perfect match; the metric is equivalent to the Dice score when $d = 0$. The different d values allow us to measure reconstructions under different degrees of deformation. Let us assume the set of all vessel points is $\mathcal{P}_{target}$ in the target data and $\mathcal{P}_{pred}$ in the prediction. Given a threshold d, every point $\mathbf{p} \in \mathcal{P}_{target}$ is marked as belonging to the set TPR(d) of true positives of the reference if there is at least one point $\mathbf{u} \in \mathcal{P}_{pred}$ satisfying $distance(\mathbf{p}, \mathbf{u}) <= d$ and to the set FN(d) of false negatives otherwise. Points $\mathbf{u} \in \mathcal{P}_{pred}$ are labelled as belonging to the set TPM(d) of true positives of the tested method if there is at least one $\mathbf{p} \in \mathcal{P}_{target}$ satisfying $distance(\mathbf{u}, \mathbf{p}) <= d$ and to the set FP(d) of false positives otherwise. The computation process is illustrated in figure 2.15 and the metric $Ot(d)$ for a certain distance threshold d can then be calculated as:

$$Ot(d) = \frac{|\text{TPM}(d)| + |\text{TPR}(d)|}{|\text{TPM}(d)| + |\text{TPR}(d)| + |\text{FN}(d)| + |\text{FP}(d)|}. \tag{2.6}$$

Reconstruction Error (*reError*) Reconstruction error (*reError*) [167] is a metric that has been proposed specifically for 3D coronary artery tree reconstruction problem.

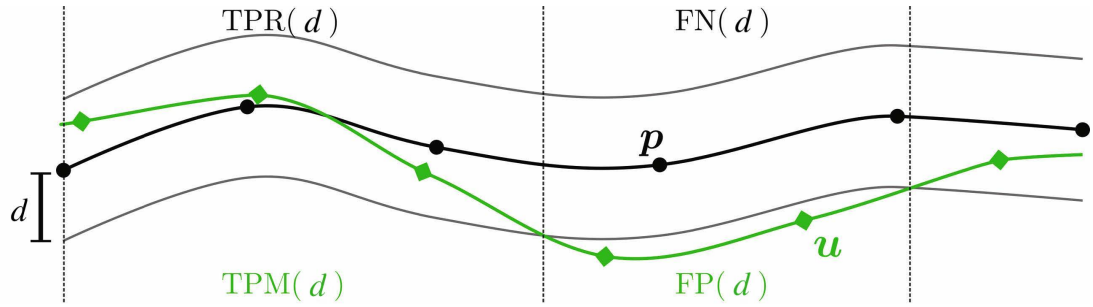


Figure 2.15: Schematic illustration of the computation of $Ot(d)$. 3D points of the test $\mathbf{u}$ (green) and ground truth $\mathbf{p}$ (black) centrelines are labelled as true and false positives or negatives depending on the sweeping test distance d. (Adapted from [202])

The binarised reconstructed volume $\widehat{\mathbf{y}}$ is compared with the original binary volume $\mathbf{y}$. *ReError* ranges from 0 to 1 where a smaller value suggests a good reconstruction result. *ReError* is calculated according to the formula:

$$reError(\widehat{\mathbf{y}}, \mathbf{y}) = 1 - \frac{|bin_{\varsigma}(\widehat{\mathbf{y}}) \cap \mathbf{y}|}{|\mathbf{y}|}. \tag{2.7}$$

Reconstruction MSE (*reMSE*) Mean squared error (*MSE*) is derived from the square of Euclidean distance that measures the average squared differences between the estimated values and the actual values. It is always a positive value that decreases as the error approaches 0. Reconstruction mean squared error (*reMSE*) between the predicted reconstruction and ground truth is computed as:

$$reMSE(\widehat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} \sum_{k=1}^{N} (bin_{\varsigma}(\widehat{\mathbf{y}})_k - \mathbf{y}_k)^2 \tag{2.8}$$

where N is the number of voxels per volume. A smaller value of *reMSE* suggests a better reconstruction.

Chamfer ℓ_2 Distance (CD_{ℓ_2}) The Chamfer distance is a widely used metric in point clouds to measure the shape dissimilarity between two sets of points. Assuming two point sets $\mathcal{S}$ and $\mathcal{S}'$, Chamfer ℓ_2 distance (CD_{ℓ_2}) is calculated based on the average sum of the Euclidean distances between nearest neighbour correspondences of the two point sets as:

$$CD_{\ell_2}(\mathcal{S}, \mathcal{S}') = \frac{1}{|\mathcal{S}|} \sum_i \min_j ||\mathcal{S}_i - \mathcal{S}'_j||_2 + \frac{1}{|\mathcal{S}'|} \sum_j \min_i ||\mathcal{S}'_j - \mathcal{S}_i||_2, \tag{2.9}$$

where i and j are the indices of all the vessel points from the two point sets, respectively, and $||\cdot||_2$ denotes the Euclidean distance or Euclidean ℓ_2 norm. For CD_{ℓ_2} measurement on voxel-based representations, the coordinates of those voxels whose values equal 1 construct the vessel point sets for predicted reconstruction after binarisation and ground truth. A smaller value of CD_{ℓ_2} represents a better result.

Earth Mover's Distance (*EMD*) Earth Mover's distance (*EMD*) [203] is a metric space measure of the difference between two frequency distributions, densities, or measurements. Informally, if the distributions are viewed as two distinct methods of piling up earth (dirt) over a metric space, the *EMD* represents the lowest cost of constructing the smaller pile with dirt removed from the larger one, where cost is calculated by multiplying the quantity of dirt moved by the distance it is moved. In comparison to CD_{ℓ_2}, *EMD* is more discriminative of the density distribution and local details. In the field of deep learning, an efficient implementation of *EMD* [204] is usually used to allow dense point cloud data. It is defined based on two point sets $\mathcal{S}$ and $\mathcal{S}'$ which have the same size:

$$EMD(\mathcal{S}, \mathcal{S}') = \min_{\phi:\mathcal{S}\to\mathcal{S}'} \sum_{\mathbf{s}\in\mathcal{S}} ||\mathbf{s} - \phi(\mathbf{s})||_2, \tag{2.10}$$

where ϕ is a bijection mapping. A smaller value of *EMD* represents a better result.

2.4.2 Statistical Analysis

Although deep learning models have achieved significant progress in a variety of tasks, their evaluations are often not supported solidly by statistical hypothesis tests. Usually, their conclusions are drawn only according to single performance scores like the mean value, or possibly aggregating more statistics such as standard deviation, which might not be sufficient to make a decision that one model outperforms another. Without enough substantial statistical tests, some declare their models modified from another attain the SOTA performance based on a sole performance score, while they actually do not improve performance. Therefore, statistical significance testing is critical in determining whether the proposed model is, in fact, improved or not.

The choice of statistical significance test is very important, as different tests selected can have different conclusions for the same evaluation. For this reason and the nature of deep learning in our work, almost stochastic order (ASO) test [205, 206] is implemented by Ulmer *et al.* [207] to compare score distributions from different models, which is designed specifically for deep learning models. ASO test builds on the concept of stochastic order by comparing the cumulative distribution functions of two models to declare one as stochastically dominant. Besides, the significance level α is an input argument to run the ASO which influences the results. ASO returns a confidence score ϵ_{min}, which indicates (an upper bound to) the amount of violation of stochastic order. In terms of analysis between model A and B using ASO, if $\epsilon_{\text{min}} < \tau$ (where the rejection threshold τ is 0.5 or less), model A is said to be stochastically dominant over model B in more cases than vice versa and model A is considered as superior. The lower ϵ_{min} it is, the more confident we can conclude that model A outperforms model B. The tests from [207] show that $\tau = 0.2$ is the most effective threshold value that has a satisfactory tradeoff between Type I and Type II error across different scenarios. Please note that for metrics such as errors where a smaller value expresses a better performance, the final confidence score ϵ_{min} should be 1 minus the returned ϵ_{min} from ASO.

2.5 Conclusion

In this chapter, we first introduce the relevant clinical background related to acquiring ICA projections. We then talk about general 3D reconstruction from a few views using deep learning in the areas of both natural images and medical images, following the introduction of several milestones during the development of deep learning. Next, we discuss 3D vessel reconstruction with an emphasis on coronary artery tree reconstruction using conventional methods and deep learning approaches. The main strength of the conventional algorithms based on mathematical methods is that they do not require a large training dataset, while deep learning methods are data-driven. However, conventional algorithms usually depend on conventional stereo vision algorithms, and usually require dense work on careful imaging

systems calibration, feature extraction, laboriously extensive key points labelling and registration, matching and fusion. Moreover, current works using mathematical methods have not been able to effectively model the cardiac or respiratory motion, while it is possible for deep learning approaches to implicitly compensate for such motions due to the strong non-linear feature mapping ability. In addition, deep learning methods can achieve faster, real-time 3D reconstructions. Finally, we describe several metrics that we use in this thesis, as well as the deep learning-specific statistical testing method.

Chapter 3

Datasets

3.1 Introduction

So far, there have been no publicly available real ICA datasets with corresponding 3D coronary artery tree data for research. The 3D coronary artery tree ground truth cannot be obtained directly from the X-ray ICA projections, even with manual intervention: it would require an additional 3D acquisition from a different modality, e.g. CCTA, on the same subjects. The lack of 3D coronary artery tree ground truth results in difficulties to learn the 3D coronary artery tree missing features via data-driven methods such as deep learning, and for direct evaluation of the reconstruction results.

3.2 Vessel Synthesis

Medical image synthesis [208, 209] is an active research area with the purpose of facilitating clinical research and evaluation when real clinical data are hard to obtain.

Some works have been proposed for generating synthetic blood vessels [210–213] based on language-theoretic models, namely, L-systems [214]. L-system [214] formalism is especially fitted to open tree structures as it has a "grammar" particularly developed for repeated branching. Therefore, its language directly supports arterial structures. Prusinkiewicz and Lindenmayer [215] extended the L-system formalism to a parametric L-system so that the arterial branching variables can be embodied into the system. Based on a parametric L-system, Zamir [210] incorporated the physiological laws of arterial branching to generate branching tree structures. The results indicate that parametric L-systems can be utilised to generate fractal tree structures, but cannot produce the variability in branching parameters found in real arterial trees. The key factor is that the bifurcation index or asymmetry ratio

α (a measure of the asymmetry of the two diameters at an arterial bifurcation, calculated as $\frac{d_2}{d_1}$, where d_1 and d_2 are the diameters of two branches) is fixed throughout one special tree bifurcation structure. This kind of uniformity is hardly observed in the physiological system and also rarely shared by arterial trees in the cardiovascular system [210]. A random value may be assigned to the value of α, but the source of variability is unknown, whether it is really random or determined by local conditions and other constraints that are hard to model. In the present case, a certain occurring probability at each bifurcation is assigned to the value of α, but so far, data from the cardiovascular system have not demonstrated any foundation for such probabilistic assignment. The values of α in the cardiovascular system could be impacted by local anatomy, local flow requirements, and other constraints, and thus the variability may not be totally arbitrary. Liu *et al.* [211] gave examples of organ blood vessels based on a stochastic parametric L-system. However, these works were limited to 2D and their properties did not resemble those measured from real CT or MRA images. Galarreta-Valverde *et al.* [212] further solved these problems to create more realistic 3D synthetic vessels via integrating stochastic and parametric rules to the L-system grammar that simulates real CT or MRA data. A public tool (namely, V-system) [213] extending L-system to generate 2D/3D synthetic vascular trees based on the grammars [212] is available on `https://github.com/psweens/V-System`.

A new synthesis approach [216] based on the space colonisation algorithm was implemented to create 3D realistic vascular structures with physiological constraints on the proliferation of branches. The synthesiser emulates the abstract behaviour of angiogenesis – a growth process in which blood vessels develop by elongating and branching from pre-existing vasculature to reach tissue devoid of any vasculature. During development, the geometry of branches and bifurcations is further constrained to replicate commonly observed vascular patterns. The code is available on `https://github.com/nikolausrauch/vessel_synthesizer`.

A vessel tree generator [171] was proposed to generate synthetic 3D RCA data informed by both clinical image data and literature values on coronary anatomy. The RCA

vessel generator is highly customisable with options in the number of branches, branch length, maximum radius, relative positions and dimensions of side branches. It also supports the definition of the number, position, and severity of stenoses. Examples of generated RCA data are illustrated in figure 3.1. The code is available on `https://github.com/kritiyer/vessel_tree_generator`.

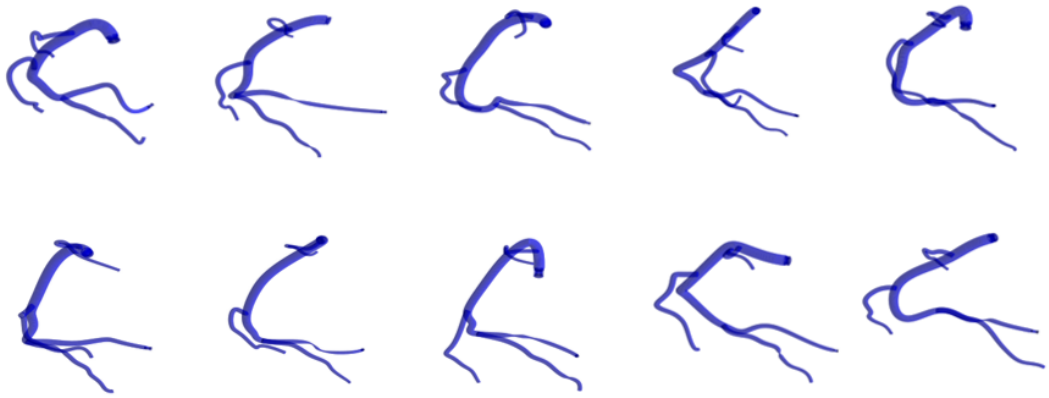


Figure 3.1: RCA data generated by vessel tree generator. (Taken from [171])

A toolbox named vessel simulator [217] was created for 2D X-ray angiographic projections synthesis to increase the number of available images as an alternative to a data augmentation method for training artificial intelligence algorithms. It has an assembly example for every vessel shape, as well as left and right coronary arteries models. The vessel simulator is able to simulate complex vessel structures, as well as stenosis and aneurysms, in X-ray angiography images. The code is available on `https://github.com/jeanschmith/vessel_simulator`.

3.3 Projection Geometry Simulation

Given a 3D coronary artery tree sample, 2D X-ray images are acquired from different views based on cone-beam forward projections. Many established software packages for projection simulation are reliable and user-friendly but inflexible, so they are not suitable for the development of advanced, efficient algorithms capable of dealing with various

geometries and constraints. Here we discuss three customisable toolboxes [218–221] for tomographic reconstruction that include projection simulation.

The All Scale Tomographic Reconstruction Antwerp (ASTRA) toolbox [218, 219] is accessible through MATLAB and Python, providing a powerful platform for algorithm prototyping, that supports 2D parallel and fan beam geometries, and 3D parallel and cone beam. Its basic forward and backwards projection operations are GPU-accelerated. The ASTRA toolbox offers a collection of building blocks that can effectively handle a range of restrictions and different geometrical characteristics of the acquisition model.

The Tomographic Iterative GPU-based Reconstruction (TIGRE) toolbox [220] offers MATLAB and Python libraries for high-performance X-ray absorption tomographic reconstruction, including a wide variety of iterative algorithms as well as FDK. The toolbox allows flexible CT geometry, including the support of a dual-axis rotational single-plane C-arm imaging system, and the geometric parameters are defined per projection, not per scan. It offers SOTA implementations of backprojection and projection operations on GPUs (including multi-GPUs) and a straightforward interface that uses higher-level languages to make it easier to create new techniques. In addition, TIGRE projectors can be integrated into PyTorch models and treated as a linear differentiable operator by the automated differentiation engine with the latest TIGRE toolbox's optional PyTorch binding.

Operator Discretization Library (ODL) [221] is a Python library that enables research in inverse problems on real or realistic data. With the help of framework, a physical model can be transformed into an operator that can be utilised in optimisation methods, for example, as a mathematical object. Moreover, without compromising speed, ODL makes it simple to test reconstruction approaches and optimisation algorithms for variational regularisation. Its differentiable projector component makes it easily fit deep learning architecture.

The setup of the ASTRA toolbox in low-level programming languages makes it less suited

to work with when creating new algorithms. In contrast, the projectors in the TIGRE [220] and ODL [221] toolboxes are capable of being integrated into existing deep learning models, because their forward-projection components are differentiable. This makes it possible to train an end-to-end deep learning model from projection domain to the image domain.

3.4 Available Datasets

There are only a few publicly available datasets for coronary arteries. Some works [222–225] only provide 3D coronary artery geometries, while some provide both projection data and corresponding 3D ground truth [157, 202, 226]. However, the paired data provided in those works [157, 202, 226] are very limited, which are mostly based on the same 4D Extended Cardiac-torso (XCAT) Phantom [227]. In addition, there are several public datasets [228–231] containing only 2D ICA data and corresponding segmentations without 3D geometry and projection information.

3.4.1 Public 3D Coronary Artery Tree Datasets

The first large-scale publicly available dataset for coronary artery segmentation, ImageCAS [222], contains 3D CCTA images of $1,000$ patients and corresponding segmented labels of both RCA and LAD. The high-dose CCTA was performed on a Siemens 128-slice dual-source scanner. To get the clearest coronary artery images, the 30–40% or 60–70% phases were chosen during the reconstruction, representing the systole or diastole phases in a cardiac cycle when the motion effect is minimal. The data were gathered from clinical cases at the Guangdong Provincial People's Hospital from April 2012 to December 2018. Inclusion was restricted to individuals who were at least eighteen years of age and had a documented medical history of peripheral artery disease, ischemic stroke, or transient ischemic attack. There were a total of 414 females and 586 males included, the average ages being 59.98 and 57.68, respectively. The captured CCTA images have sizes of $512 \times 512 \times (206 - 275)$ voxels, a planar resolution of 0.29–0.43 mm^2, and spacing

of 0.25–0.45 *mm*. Two radiologists independently labelled the RCA and LAD in each CCTA image, and their results were cross-validated. In case of discrepancy, an extra radiologist did the annotation and the final result was decided by consensus. An example of ImageCAS data is illustrated in figure 3.2.

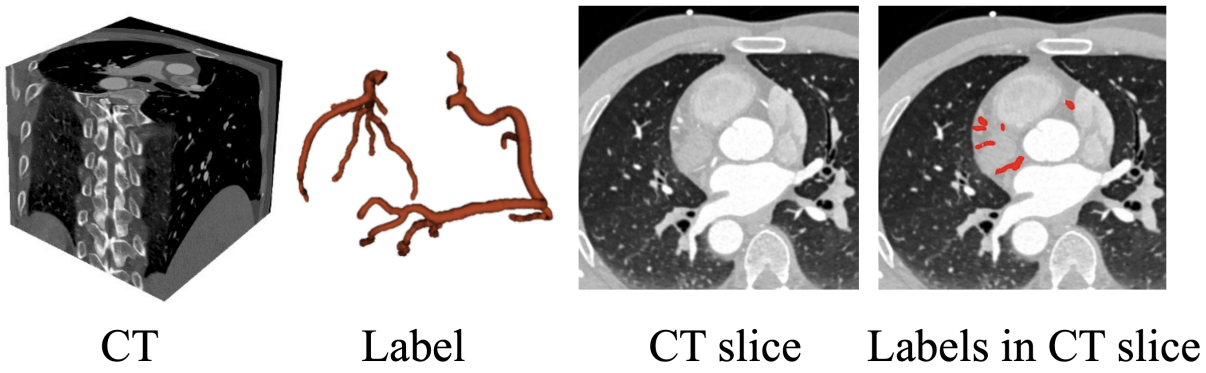


Figure 3.2: An example of ImageCAS data. (Taken from [222])

The ASOCA (automated segmentation of normal and diseased coronary arteries) challenge presents an anonymised CCTA dataset [223, 224] that includes voxel-wise annotations and related data in the form of centrelines, calcification scores, and coronary lumen meshes in 20 normal and 20 diseased cases. The ASOCA challenge test set consists of 10 normal and 10 diseased CCTA instances that lack publicly accessible ground truth annotation and other related data. 30 individuals with documented coronary disease and 30 patients without any disease reported by the cardiologist were included in the selection, which was stratified based on accessible medical records. The CCTA data was gathered by the University of New South Wales using a GE LightSpeed 64 slice CT scanner from Mercy Angiography, Auckland. In particular, retrospective ECG-gated acquisition was used to gather the data, and the late diastole time point was chosen for reconstruction. Three specialists worked independently to complete high-quality voxel-wise segmentations. The segmentation process began with a thresholding operation at a threshold selected by the expert, and then the vessel outlines and over- and under-segmented vessels were manually corrected. Using majority voting, the three annotators' segments were combined; that is, pixels that were identified as coronary vessels by at least two annotators were added to the vessel mask to produce the final annotation. In order to link vessel segments that

were detached from the coronary tree because of excessive annotator variance, especially at strongly calcified locations, the vessel mask was post-processed to remove thin vessels that would not be clinically meaningful. Anisotropic resolution is present in both the gathered CCTA images and annotations, with an out-of-plane resolution of 0.625 mm and an in-plane resolution of 0.3–0.4 mm, depending on the patient. Finally, to capture the artery shape, smooth surface models with an average of $37,000$ vertices were created from the voxel annotations.

A dataset, namely USCT3D, contains 29 3D computational models of arterial trees, which were created from a sample [225] of 20 patients with known or suspected stable coronary disease, 20 of which came from intravascular ultrasound (IVUS) and 9 from CCTA, based on Hospital das Clínicas Medical School of the University of São Paulo. Five patients have both CCTA and IVUS data available. Two different scanners, a 64-row scanner (Aquilion 64, Toshiba Medical Systems, Otawara, Japan) and a 320-row scanning system (Aquilion ONE, Toshiba Medical Systems, Otawara, Japan), provided the CCTA images. To minimise radiation exposure, all captures were prospectively initiated at 75% of the cardiac cycle. After segmenting the obtained CCTA images, a marching cubes technique was used to define the lumen. With the AtlantisTM SR Pro Imaging Catheter 40 MHz ECG-triggered and attached to an iLabTM Ultrasound Imaging System, the IVUS pictures were obtained. These images were gated to retrieve the diastolic cardiac phase. A professional manually divided the lumen area using cubic splines. After being acquired from either the CCTA or IVUS imaging modalities, these meshes underwent further processing to produce volume meshes for 3D simulations. Finally, the 9 CCTA geometries generated 8 LAD artery, 3 left circumflex (LCx) artery, and 1 ramus intermedius (RI) meshes, and the 20 IVUS geometries generated 15 LAD, 2 LCx, and 3 RCA meshes.

Apart from the aforementioned three clinical datasets [222–225] providing 3D coronary artery geometries of real patients, there are some works [157, 202, 226] using 4D XCAT software [227] for synthetic simulation.

A dataset named Vukicevic-CARecon consists of three different subdatasets [157] providing

digital phantom, static phantom, and clinical data. Subdataset I is a realistic digital phantom created using the 4D XCAT program [227]. They adjusted the frame rate to 30 frames per second, cardiac and respiratory cycles to 1 and 5 seconds, respectively, and image resolution to 960×960 pixels with a pixel spacing of 0.184 mm. Subdataset II is a physical static phantom composed of interconnected rubber tubing that contains the iodinated contrast agent. Using a flat-panel detector (1024×1024 pixel resolution at 15 frames per second and a pixel size of 0.287 mm), the data were acquired by the GE Innova 2100-IQ angiography system. To get the phantom's reference geometry, a Toshiba Aquilion multi-slice 64-slice CT system was used. A total of 467 slices were obtained, each having a thickness of 0.35 mm and a resolution of 512×512 pixels. The subdataset II thus gives paired data of 2D angiograms and corresponding 3D geometry. In subdataset III, two coronary arteries, one single branch and one complete LCA, routinely collected during patient examinations at the Clinical Centre Kragujevac in Serbia, are made publicly available. A routine protocol and the same X-ray imaging equipment utilised for the phantom subdataset II were employed for all acquisitions. Only 2D ICA projections are given for the LCA, whereas both 3D geometry and 2D angiograms are provided for the single-branch data.

Based on the framework [226], the coronary artery reconstruction challenge (CoronARe) [202] leveraged the LCA geometry of 4D XCAT Phantom [227] which defines detailed and anatomically correct cardiac vasculature using NURBS and allows for the simulation of cardiac motion, to simulate projection data for both symbolic and tomographic reconstruction. Throughout the acquisition period, the authors [202] simulated a series of 133 NURBS descriptions of the LCA, with the heart rate set to 80 beats per minute. To get 3D ground truth for the symbolic reconstruction, the spline defining the centrelines was sampled at regular intervals of 0.3 mm in arc length. The 3D spline was projected onto the appropriate image plane for every projection in the series. These splines were sampled from points defining the uncorrupted centreline segmentations at regular arc length intervals of 2.0 mm, much like in 3D. The NURBS files were also voxelised with an isotropic image

spacing of 0.3 mm. The values of the voxels that correspond to the locations of arteries were set to one, while the values of the remaining voxels were set to zero. The ground truth for each time step in the sequence was defined by a $512 \times 512 \times 360$ subvolume centred at the barycentre of the 3D ground truth points of the end-diastolic phase. The projection images were simulated using the CT Projector [227], which can analytically calculate the sum of attenuation values from NURBS specifications given the imaging geometry. The forward projection obeys a standard rotational angiography protocol. Specifically, a single 5.3-second sweep on a circular source trajectory spanning 200° yielded 133 photos. The projection images have an isotropic size of 0.32 mm and are 960×960 pixels in both the horizontal and vertical directions. The source-to-isocentre and source-to-detector distances are 800 mm and 1200 mm, respectively. Both the symbolic and tomographic projection data sets were subjected to random corruption of the acquisition, which increases in severity.

All the aforementioned public 3D coronary artery tree datasets are summarised in table 3.1.

Table 3.1: Summary of public 3D coronary artery tree datasets.

Dataset Name		Collection Center	Modality	Size	Representation	Resolution	Corresponding 2D Provided?	Annotation (3D Segmented?)
ImageCAS [222]		The Guangdong Provincial People's Hospital	CCTA	1,000	Voxel	$512 \times 512 \times (206 - 275)$	No	Yes
ASOCA [223, 224]		Mercy Angiography, Auckland	CCTA	60	Mesh	N/A	No	Yes (40 out of 60)
USCT3D [225]		Hospital das Clínicas Medical School of the University of São Paulo	CCTA/IVUS	29	Mesh	N/A	No	Yes
Vukicevic - CARecon [157]	I	N/A	Realistic Digital Phantom	1	Mesh	N/A	Yes	Yes
	II	N/A	Physical Static Phantom	1	Mesh/Voxel	$512 \times 512 \times 467$	Yes	Yes
	III	Clinical Center Kragujevac, Serbia	CCTA	1	Mesh	N/A	Yes	Yes
CoronARe [202]		N/A	Realistic Digital Phantom	1	Voxel	$512 \times 512 \times 360$	Yes	Yes

3.4.2 Public 2D ICA Datasets

There are several public datasets containing only 2D ICA images and their segmentation annotations without corresponding 3D geometry and projection information, as concluded in table 3.2. (1) The X-ray angiography coronary vessel segmentation dataset [228] comprises 126 coronary angiograms annotated by expert radiologists in the testing set,

and $1,621$ mask frames and $1,621$ coronary angiograms in the training set. (2) There are two instance segmentation tasks in the Automatic Region-based Coronary Artery Disease Diagnostics using X-ray angiography pictures [229] dataset. $1,500$ pictures of coronary artery trees and their corresponding comments are included in the first task. A separate set of $1,500$ pictures with areas having atherosclerotic plaques labelled is included in the second challenge. (3) A dataset [230] includes 120 sequences of real clinical X-ray coronary angiography images that were obtained from Ren Ji Hospital of Shanghai Jiao Tong University. Each sequence is between 30 and 140 frames long. To create the ground truth, three professionals manually annotated 323 images from the 120 sequences. (4) There are 130 X-ray coronary angiograms in the database in [231], together with the ground-truth image that corresponds to each one, which has been outlined by a qualified cardiologist.

Table 3.2: Summary of public datasets of 2D X-ray coronary angiograms.

Dataset Name	Collection Center	Imaging Device	Size	Resolution	Annotation? (segmentation)
XCAD [228]	Not specified	General Electric Innova IGS 520 system	1747	512×512	Yes
ARCADE [229]	The Research Institute of Cardiology and Internal Diseases, Almaty, Kazakhstan	The Philips Azurion 3 and the Siemens Artis Zee	3000	512×512	Yes
XCA [230]	Renji Hospital of Shanghai Jiao Tong University	800 mAh digital silhouette angiography X-ray machine from Siemens and medical angiography X-ray system from Philips	323	512×512	Yes
DCA1 [231]	The Cardiology Department of the Mexican Social Security Institute, UMAE T1-León	Not specified	130	300×300	Yes

3.4.3 Private Clinical Datasets

We collected our own 2D clinical ICA dataset with corresponding projection geometry information recorded. The clinical ICA dataset is being acquired as part of a substudy of the Oxford Acute Myocardial Infarction (OxAMI) Study, investigated by Dr. Abhirup Banerjee and Prof. Robin Choudhury. The OXAMI study is a prospective single centre observational cohort study of patients with suspected or known coronary artery disease, who present to the John Radcliffe Hospital in Oxford and undergo coronary angiography with the possibility of proceeding to PCI. The South Central – Oxford C Research Ethics Committee has reviewed and approved the study, and the study is sponsored by the University of Oxford and the NIHR Oxford Biomedical Research Centre (Ethics Ref:

11/SC0397). The data collected from the study are recorded anonymously and individual patients will not be identified where both personal and clinical details of the patients remain strictly confidential.

The clinical dataset used in the thesis contains ICA images of 8 patients, which were acquired based on Siemens Artis Interventional Angiography Systems with provided informed consent. These 2D angiograms consist of multiple cardiac cycles (at least three), ranging from 16 to 64 temporal frames. For RCA, the angiogram images are typically acquired by varying the primary angle at LAO (25 to 35°) and AP (−5 to 5°) views and the secondary angle at cranial (25 to 35°), straight (−5 to 5°), and, in a subset of patients, caudal (−35 to −25°) views. For LAD artery, the angiograms are typically acquired by varying the primary angle at LAO (25 to 35°), AP (−5 to 5°), and RAO (−35 to −25°) views and the secondary angle at cranial (25 to 35°) and straight (−5 to 5°) views. For the LCx artery, the secondary angle usually varies between straight (−5 to 5°) and caudal (−35 to −25°) views. The manual segmentations of the coronary artery structures from 2D angiographic projections were annotated by trained experts on a proprietary graphical user interface tool developed in our research group [154]. In total, for this project, we acquired RCA data of 8 patients, where two of the patients contained three ICA projections, while the rest contained two, and LAD data of 3 patients, where one of the patients contained three ICA projections, while the rest contained two. Figure 3.3 shows two ICA samples from our clinical dataset for both RCA and LAD and their corresponding manual coronary arteries segmentation results based on a standard single-plane X-ray angiography system.

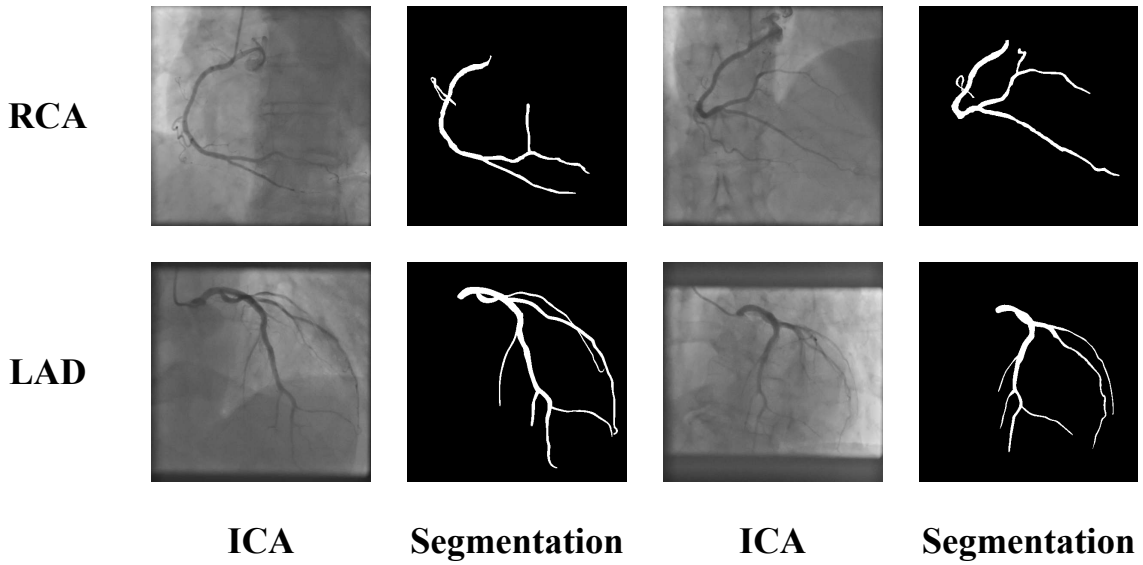


Figure 3.3: Two ICA images of both RCA and LAD (top to bottom) and the corresponding coronary arteries segmentation results (left to right) based on a standard single-plane X-ray angiography system. The two RCA angiograms are acquired at LAO 29.8° and 0.4° (LAO-Straight view), and at −3.6° and CRA 36.6° (AP-CRA view), in terms of primary and secondary angles. The two LAD angiograms are acquired at −0.1° and CRA 32.7° (AP-CRA view), and at LAO 30.6° and CRA 21.6° (LAO-CRA view), in terms of primary and secondary angles.

3.5 Conclusion

To sum up this chapter, we present different methods of vessel synthesis to augment datasets as well as show the current available public datasets for 3D coronary artery tree and ICA data. We describe a private clinical ICA dataset we collected. We also compare different tools for 2D projection simulation from 3D data.

Specifically in the thesis, ImageCAS [222] is used as the main dataset across all four contributions in chapters 4 to 7 as it has a number of 3D segmented CCTA data which contain coronary tree geometries of real patients. The original ImageCAS dataset contains $1,000$ data with both RCA and LAD in the same 3D space, so we apply connected component analysis [232] to separate them. We then screen the separated data and keep 879 RCA data with correct separation and complete vasculature for use in chapter 5. We further screen the 879 3D RCA data and drop the ones whose simulated 2D projections are mostly out of the projection plane boundary after applying severe rigid transformations, so there are remaining 669 RCA samples in total for use in chapter 6. At the same time, we

use these 669 RCA samples to complete chapter 4 (the ImageCAS dataset came out after we had implemented our model in chapter 4). Finally, we reprocess the original ImageCAS to create a different point cloud-based dataset for use in chapter 7, where we use 810 point cloud-based anatomies. Moreover, to prove the proposed model's generalisability in chapter 6, we additionally use the ASOCA dataset [223, 224] (also termed as CCTA-UNSW dataset) and a synthetic dataset generated by the vessel tree generator [171] for testing from the unseen domains. We do not continue using these two unseen datasets in chapter 7 as the proposed method in chapter 7 has not yet achieved very satisfactory results in the source domain. The private clinical dataset has been used across all the last three contributions in chapters 5 to 7, except the first contribution in chapter 4, where we do not consider real clinical scenarios.

Chapter 4

Neural Implicit Representation

Abstract - NeCA: 3D Coronary Artery Tree Reconstruction from Two 2D Projections via Neural Implicit Representation

This chapter explores the feasibility of 3D coronary artery tree reconstruction from only two projections without any motion consideration between projection planes. A self-supervised deep learning method called NeCA is proposed, which is based on neural implicit representation using the multiresolution hash encoder and differentiable cone-beam forward projector layer, in order to achieve 3D coronary artery tree reconstruction from two 2D projections. The proposed NeCA method is validated using six different metrics on a dataset generated from CCTA of RCA and LAD artery. The evaluation results demonstrate that the NeCA method, without requiring 3D ground truth for supervision or large datasets for training, achieves promising performance in both vessel topology and branch-connectivity preservation compared to the supervised deep learning model. The code to this work is available at: `https://github.com/WangStephen/NeCA`.

Publication: Wang, Y., Banerjee, A., & Grau, V. (2024). NeCA: 3D Coronary Artery Tree Reconstruction from Two 2D Projections via Neural Implicit Representation. Bioengineering, 11(12), 1227.

4.1 Introduction

The emergence and prosperity of deep neural networks have enabled 3D automated reconstruction from limited views in medical images. Most of them need large training datasets and work in a supervised learning manner, but the acquisition of paired data has always been a challenge in real clinics. Recently, NeRFs [40] have made a significant

contribution to the field of computer vision, allowing for neural implicit representation and novel view synthesis. In neural implicit representation learning, a bounded scene is parameterised by a neural network as a continuous function that maps spatial coordinates to metrics such as occupancy and colour. The optimization of NeRFs only relies on several images from different viewpoints. Based on NeRFs, neural attenuation field (NAF) [144] was proposed to tackle the problem of sparse-view cone-beam CT reconstruction, which require at least 50 projections. Shen *et al.* [145] proposed a neural implicit representation learning methodology to reconstruct CT images, which performs on 10, 20, and 30 projections.

Some deep learning-based studies attempted 3D coronary artery tree reconstruction from limited projections. Wang *et al.* [168] used CCTA data to simulate projections and trained a weakly supervised adversarial learning model for 3D reconstruction from two projections. However, their model requires large training datasets ($8,800$ data in the experiments), with the 3D ground truth used in the discriminator. In [169–171], 3D synthetic coronary artery tree data and simulated corresponding 2D projections were generated to train supervised learning models; their models require more than two projections for training. Bransby *et al.* [172] used bi-planar ICA data to reconstruct a single coronary tree branch in a supervised learning setup. Maas *et al.* [173] proposed a NeRFs-based model to achieve 3D coronary artery tree reconstruction from limited projections without involving 3D ground truth in training. However, they tested the performance only on two 3D studies, and the number of required projections is at least 4. Despite the improvement in deep neural networks, 3D coronary artery tree reconstruction from two projections without involving corresponding 3D ground truth and large training datasets remains challenging.

In this chapter, we propose a self-supervised deep learning method named NeCA, which is based on neural implicit representation to achieve 3D coronary artery tree reconstruction from only two projections. Our method requires neither 3D ground truth for supervision nor large training datasets. It iteratively optimises the reconstruction results in a self-supervised fashion with only the projection data of one subject as input. Our proposed

method utilises the advantages of the multiresolution hash encoder [99] to encode point coordinates, residual MLP to predict point occupancy, and a differentiable cone-beam forward projector layer [221] to simulate projections. The simulated projections are then learned from the input projections by minimising the projection error in a self-supervised manner. Our method aims to learn and optimise the neural representation for the entire image and can directly reconstruct the target image by incorporating the forward model of the imaging system. We use a public CCTA dataset [222] to validate our model's feasibility on the task based on six metrics. The evaluation results indicate that our proposed NeCA model, without 3D ground truth for supervision or large datasets for training, achieves promising performance in both vessel topology preservation and maintaining branch connectivity compared to an equivalent supervised learning model. The main contributions of this work are:

1. **3D coronary tree reconstruction using self-supervised learning from only two projections:** Our proposed deep learning method achieves 3D coronary artery tree reconstruction from two projections where neither 3D ground truth for supervision nor large training datasets are required.

2. **Neural implicit representation learning:** We leverage the advantages of MLP neural networks as a continuous function to represent the coronary tree in 3D space in order to enable mapping from encoded coordinates to corresponding occupancies.

3. **The applications of multiresolution hash encoder and differentiable cone-beam forward projector layer:** We combine a learnable hash encoder and a differentiable projector layer in our model to allow for efficient feature encoding and self-supervised learning from 2D input projections.

4. **Evaluations:** We perform thorough evaluation of our model on the RCA and LAD artery in terms of six quantitative metrics.

4.2 Materials and Methods

4.2.1 Dataset

We use a public CCTA dataset [222] containing binary segmented coronary artery trees for our study, splitting the coronary artery trees into the RCA and LAD artery. Since our model is an optimization-based method for each individual data point, we do not need training/validation split. We use 67 RCA data and 79 LAD data points as the test set. We perform cone-beam forward projections on the CCTA data to generate the input projections with simulated attenuated X-ray intensities based on the ODL [221]. For each CCTA data point, we generate only two projections to use in our model for 3D coronary artery tree reconstruction. The projection geometries for RCA and LAD are illustrated in table 4.1, which mimic the ones generally used in clinics. Figure 4.1 illustrates an example of two projections generated from both RCA and LAD.

Table 4.1: Projection geometry to simulate cone-beam forward projections for both RCA and LAD.

Data	Geometry	First Projection Plane	Second Projection Plane
RCA and LAD	Detector spacing	0.2769×0.2769 mm^2 to 0.2789×0.2789 mm^2	
	Detector size	512×512	
	Volume spacing	$90 \times 90 \times 90$ mm^3 to $105 \times 105 \times 105$ mm^3	
	Volume size	$128 \times 128 \times 128$	
RCA	DSD	970 mm to 1010 mm	1050 mm to 1070 mm
	DSO	745 mm to 785 mm	$\pm$3 mm to the 1st projection
	Primary angle	18° to 42°	$-8°$ to 8°
	Secondary angle	$-8°$ to 8°	18° to 42°
LAD	DSD	1030 mm to 1090 mm	+70 mm to the 1st projection
	DSO	740 mm to 760 mm	+3 mm to the 1st projection
	Primary angle	$-8°$ to 8°	$-47°$ to $-23°$
	Secondary angle	18° to 42°	21° to 45°

Figure 4.1: An example of two projections generated from RCA and LAD data.

4.2.2 Proposed Model

Our proposed model NeCA consists of five stages and allows for end-to-end learning. First, we normalise the coordinate index in the image spatial field according to resolution. Then, for each voxel point, we use a multiresolution hash encoder [99] to encode their normalised coordinates to obtain the corresponding multiresolution spatial feature vectors. These feature vectors are next sent to the residual MLP to predict the occupancy at the position of that point. The occupancy predictions of all the points form the 3D coronary artery tree reconstruction results. After that, we simulate the X-ray forward projections from the 3D predicted reconstruction based on the projection geometry of the input. Finally, these simulated projections are learned iteratively against the input projections in a self-supervised way. Stages 2 to 5 of our proposed model are illustrated in figure 4.2.

Coordinate Normalisation

The input to the model is a set of integer coordinates $\mathbf{x} = (x, y, z)$ based on the number of voxels $n_{vx} \times n_{vy} \times n_{vz}$ in 3D volume ranging in (1 to n_{vx}, 1 to n_{vy}, 1 to n_{vz}). We normalise the coordinates from these voxels according to the voxel spacing $s_{vx,vy,vz}$ along each axis, as calculated in equation (4.1). These normalised coordinates $\mathbf{x}' = (x', y', z')$ are then sent to a multiresolution hash encoder at the next stage to efficiently obtain the corresponding

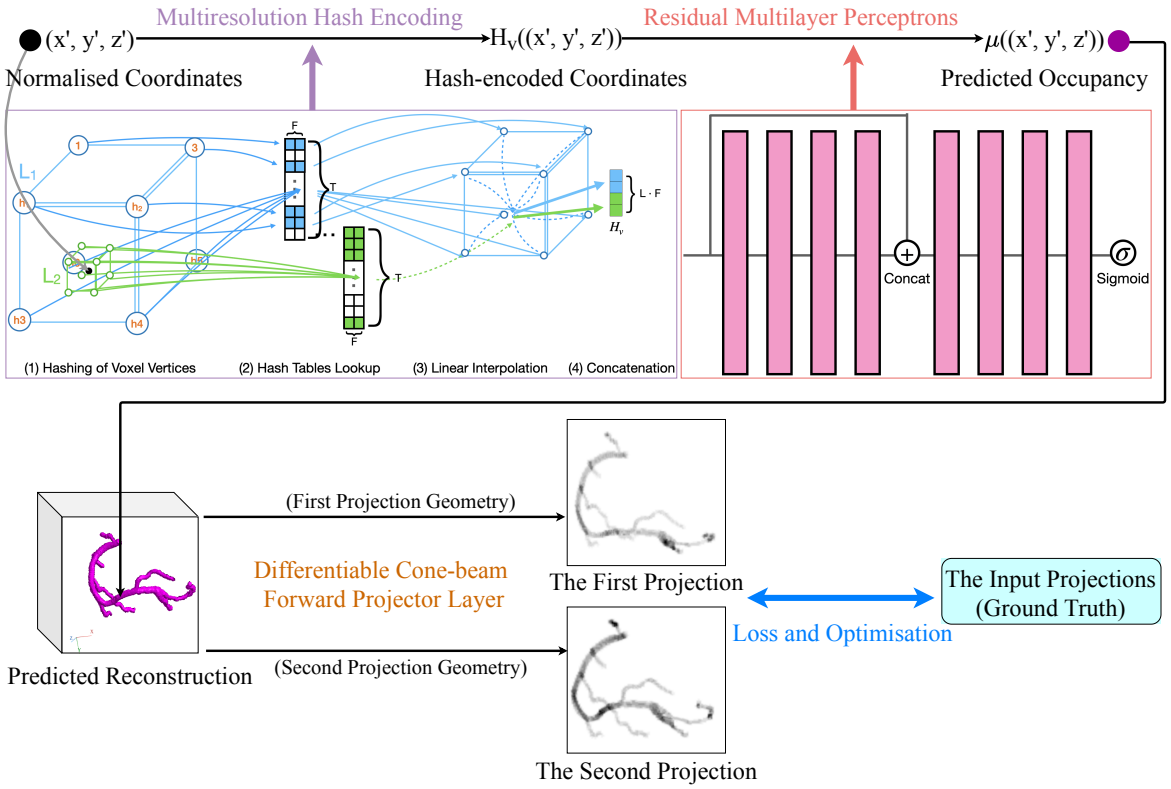


Figure 4.2: Stages 2 to 5 of our proposed NeCA model. **1.** The normalised coordinates (x', y', z') of the sampled point are input to the multiresolution hash encoder to obtain the corresponding feature vectors $H_v((x', y', z'))$. The multiresolution hash encoder in this figure shows an example of 2 resolution levels coloured by green and blue from the fine to coarse resolution for one sampled point in black. (1.1) Hashing of voxel vertices: we apply L resolution levels (L_1 and L_2 in this figure) shaped as grid cubes for the black point and we hash the coordinates of the vertices of the same resolution cube. (1.2) Hash tables lookup: according to the resulting hash values $(h_{1,2,...8})$ on all the vertices of each resolution level, we find the corresponding F-dimensional feature vectors from the learnable hash table with size T of that resolution level. (1.3) Linear interpolation: we linearly interpolate these feature vectors on the vertices based on their relative positions to the sampled point at each resolution level. (1.4) Concatenation: we concatenate the interpolation results of each resolution level to obtain the final multiresolution feature vectors $H_v((x', y', z'))$ with size $L \times F$ as the hash-encoded coordinates for the sample point. **2.** The hash-encoded coordinates $H_v((x', y', z'))$ of the sample point are then mapped to corresponding predicted occupancy of coronary tree point $\mu((x', y', z'))$ by residual MLP. **3.** We next perform differentiable forward projections on the 3D predicted coronary tree using the same projection geometry as the input projections to generate two predicted projections. **4.** The two predicted projections are finally learned and optimised from the input projections in a self-supervised learning way.

spatial feature vectors.

$$n'_{x,y,z} = \frac{n_{vx,vy,vz} \times s_{vx,vy,vz} - s_{vx,vy,vz}}{2},$$

$$\mathbf{x}' = \text{Norm}((x, y, z)) = (-n'_x + (x-1) \times s_{vx}, -n'_y + (y-1) \times s_{vy}, -n'_z + (z-1) \times s_{vz}). \tag{4.1}$$

Multiresolution Hash Encoding

We use the multiresolution hash encoder [99] $H_v = enc(\mathbf{x}'; \mathbf{\Theta})$ to encode the normalised positions of sampled points, which enables fast encoding without sacrificing performance. With the multiresolution structure, it allows the encoder to disambiguate hash collisions. The multiple resolutions are arranged into L levels with different T-dimensional learnable hash tables at each level containing feature vectors with size F. The hyperparameters of our multiresolution hash encoder are shown in table 4.2, and the structure of the encoder is illustrated in figure 4.2.

Table 4.2: Hyperparameters for the multiresolution hash encoder used in our work.

Parameter	Symbol	Value
Number of levels	L	16
Maximum entries per level (hash table size)	T	2^{19}
Number of feature dimensions per entry	F	2
Coarsest resolution	N_{min}	16
Resolution growth factor	b	2
Input dimension	d	3

For each voxel, we apply L resolution levels, which are independent of each other. The resolution size N is chosen based on an exponential increment between the coarsest and finest resolutions $\lfloor N_{min}, N_{max} \rfloor$, where N_{max} is selected to match the finest detail in the training data. It is defined as:

$$N_l \coloneqq \left\lfloor N_{min} * b^l \right\rfloor, \tag{4.2}$$

where $l \in \{0, 1, \dots L-1\}$, and $b = 2$ is the growth factor. For a single level N_l, the input point with normalised coordinates $\mathbf{x}' = (x', y', z') \in \mathbb{R}^3$ is geometrically scaled to a grid cube containing 2^3 vertices according to the grid resolution at this level. To implement this functionality, the original 3D volume is evenly split into a number of grid cubes according to the resolution N_l^3, and the grid cube containing the desired sampled point is

assigned to this point as the spanned grid cube. The multiresolution property in the hash encoder covers the full range from the coarsest resolution N_{min} to the finest resolution N_{max}, which ensures that all scales are contained, in spite of sparsity. The four parts of the multiresolution hash encoder are discussed in detail below.

Hashing of Voxel Vertices For all normalised voxels after scaling at resolution level N_l, we have $(N_l + 1)^d$ vertices in total. For coarse levels when $(N_l + 1)^d <= T$, we have one-to-one mapping from all the vertices at this resolution level N_l to hash table entries, so there is no collision. Regarding finer levels when $(N_l + 1)^d > T$, we use a hash function h to index into the feature vector array, effectively treating it as a hash table. In this case, we do not explicitly tackle hash collisions, but instead we reply on gradient-based optimisation in the backpropagation of the subsequent residual MLP to automatically handle them. For instance, if two voxels have the same hash value on one or more vertices, the voxel closer to the desired object which our model is more focused on tends to have larger gradients during optimisation, so this voxel takes the domination to update the collided feature vector entry. In this way, the collision issue is handled implicitly.

We assign indices to these vertices by hashing their coordinates. The spatial hash function [233] h is defined in the following form:

$$h(\mathbf{x}') = (\oplus_{i=1,2,3} \mathbf{x}'_i \pi_i) \mod T \tag{4.3}$$

where $\mathbf{x}'$ is the input point, $\mathbf{x}'_{i=1,2,3}$ are the corresponding spatial normalised coordinate values, $\oplus$ denotes the bit-wise XOR operation, π_i are unique large primary numbers, and T is the hash table size.

Hash Tables Lookup We now have the hash value for each vertex at each resolution level of each point. We then maintain an individual learnable hash table, which contains T numbers of F-dimensional feature vectors for each resolution level. For the hash values on all the vertices of each resolution level, we look up the corresponding entries in the level's respective feature vector array, i.e., the hash table. Next, the previously assigned indices

on the vertices are replaced by the corresponding lookup feature vectors, so each resolution level conceptually stores feature vectors at the vertices of a grid cube. The hash tables at different resolution levels are the only trainable parameters $\mathbf{\Theta}$ in the multiresolution hash encoder, and the size of these parameters is $L \times T \times F$.

Linear Interpolation For each resolution level, we linearly interpolate the feature vectors on the vertices according to their relative positions to the sampled point within this resolution level cube. Interpolating the queried hash table entries guarantees the encoded feature vectors with the later residual MLP are continuous during network training. After interpolation, the final feature vectors with the dimension F for the sampled voxel at this resolution level are produced.

Concatenation We concatenate the interpolated feature vectors for each resolution level to generate the final multiresolution hash encoding feature vectors $H_v \in \mathbb{R}^{L \times F}$ for the sampled point, which can then be utilised to predict the occupancy of coronary artery tree for this point position by the residual MLP at the next stage. The dimension $L \cdot F$ for the final encoded feature vectors of each voxel is regarded as the channel dimension for later residual MLP training.

Residual MLP

We exploit residual [37] MLP $m(H_v; \mathbf{\Phi})$ to predict the occupancy value μ from the position-encoded feature vectors H_v of each point, where $\mathbf{\Phi}$ is the trainable weight parameters of the residual MLP. The residual MLP network serves as a continuous function to implicitly parameterise a bounded scene, i.e., the 3D coronary artery tree in our case, which maps spatial coordinate features to the predicted occupancy values. This, in fact, encodes the internal information of an entire 3D coronary tree into the network parameters.

The residual MLP contains eight fully connected layers, as depicted in figure 4.2. We apply residual learning in the middle layer to preserve the original feature information. The residual MLP receives the feature vectors as input with $L \cdot F$-dimensional channels

and produces predicted occupancy values with a 1-dimensional channel. The feature dimensions for all the hidden layers are 256-wide. Except for the last layer followed by a sigmoid activation, all the layer outputs are followed by LeakyReLU activation [234].

Differentiable Forward Projector Layer

At this stage, we have all the predicted occupancy values for all the voxels, which construct the 3D coronary artery tree reconstruction results. After that, we simulate the X-ray cone-beam forward projections from the 3D reconstruction results based on the same projection geometry as the input projections to generate two predicted projections. The forward projection simulation is based on the theory that the intensity of an X-ray beam is reduced by the exponential integration of attenuation coefficients along the ray path. We use ODL [221] to implement this differentiable X-ray forward projector layer that enables self-supervised loss optimisation at the final stage.

Loss

We use *MSE* loss to calculate the differences between the input projections and simulated forward projections. The loss function $\mathcal{L}$ is defined as follows:

$$\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}) = \frac{1}{N \times I} \sum_{n=1}^{N} \sum_{i=1}^{I} (P_{ni} - G_{ni})^2 \tag{4.4}$$

where N (= 2 in our work) is the number of projections, I (= 512^2 in our work) is the number of pixels in one projection, P is the simulated projection, and G is the corresponding input projection.

The loss function is used to learn the multiresolution hash tables $\mathbf{\Theta}$ and the residual MLP $\mathbf{\Phi}$ during training. With this, the 3D occupancy predictions are improved iteratively based on the optimisation of 2D projection errors. After training, the final 3D coronary artery tree can be rendered with the predicted occupancy values, after binarisation with 0.5, by querying all the voxels with their coordinates from the model.

4.2.3 Training Setup

We implement our proposed model using PyTorch [235] and choose the Adam optimiser [236] with a learning rate of 10^{-4}. The number of epochs for optimisation is $5,000$. The learning was performed on an HPC cluster utilising Nvidia Tesla v100 GPUs. The package versions we used for NeCA are Python 3.8.17, PyTorch 1.9.0, and ODL 1.0.0.dev0.

4.2.4 Baseline Model

We use the supervised learning model 3D U-Net [237] as our baseline model. We follow the original 3D U-Net architecture with three sampling levels and a bottleneck layer using the same number of convolutional filters. The channel size for both the input and output to 3D U-Net model in our work is 1, as illustrated in figure 4.3. The input to 3D U-Net is an ill-posed volume reconstructed from two clinical-angle projections of the 3D coronary tree by a conventional backprojection method, and the output is the 3D coronary artery tree reconstruction result. We train two 3D U-Net models based on the CCTA dataset [222] using 669 RCA data and 788 LAD data, respectively, where we split them into 75% training, 15% validation, and 10% test data. The test datasets here are the same datasets used for testing our proposed model.

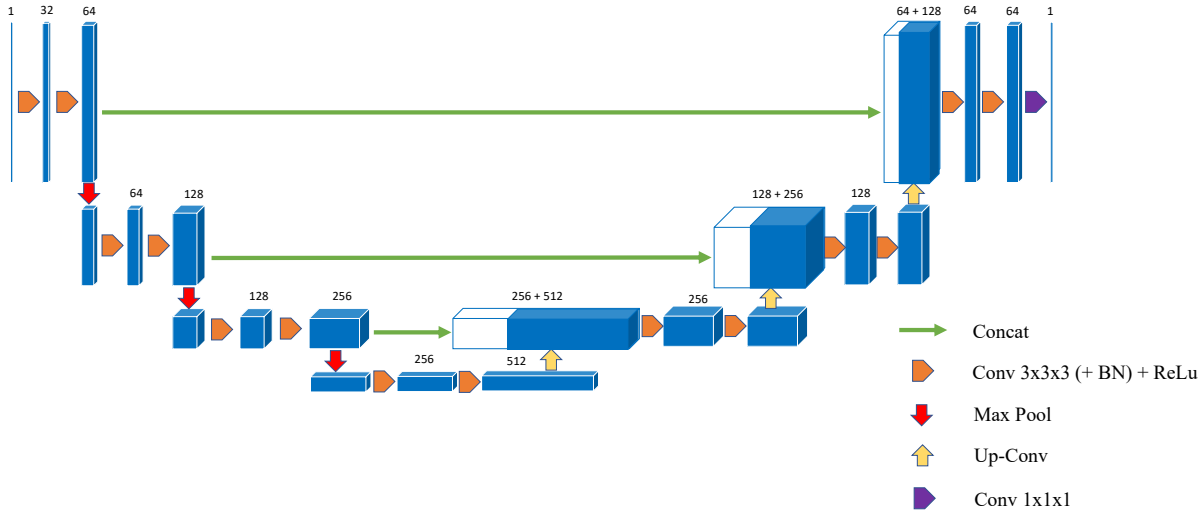


Figure 4.3: Architecture of 3D U-Net model.

We implement the 3D U-Net baseline model using PyTorch [235] and choose the Adam optimiser [236] with an initial learning rate of 10^{-4}. A learning rate decay policy is used, where the learning rate is decayed by 0.1 if no improvement is observed after 10 epochs. We use an early stopping strategy to avoid overfitting when there is no more improvement after 15 epochs. The training was performed with a batch size of 3 on an HPC cluster utilising Nvidia Tesla v100 GPUs. The models are trained with *MSE* loss.

4.2.5 Evaluation Metrics

We employ six metrics for evaluation between the 3D coronary artery tree reconstruction results and the original CCTA data (ground truth): *clDice* [201], *Dice*, *IoU*, *reError* [167], CD_{ℓ_2}, and *reMSE*. Before evaluation, we apply connected component analysis [232] on our reconstructed coronary artery tree to remove sparse disconnected objects with less than 25 voxels.

4.3 Results

We perform both quantitative and qualitative evaluations on both RCA and LAD datasets. Apart from the clinical-angle projections simulated according to table 4.1, we additionally test 3D reconstructions based on two orthogonal views using our NeCA model for comparison (termed as NeCA (90°)).

4.3.1 Quantitative Results

We quantitatively evaluate our NeCA model, NeCA (90°), and supervised 3D U-Net model on 67 RCA test data and 79 LAD test data.

RCA Dataset

Performance over Six Metrics We evaluate NeCA, NeCA (90°), and the 3D supervised U-Net model in terms of six metrics, namely *clDice*, *Dice*, *IoU*, *reError*, CD_{ℓ_2}, and *reMSE*.

The quantitative results are presented in table 4.3.

Table 4.3: Quantitative evaluation results of NeCA, NeCA (90°), and supervised 3D U-Net model on 67 RCA test data in terms of six metrics. Best results of each metric are in **bold**.

Model	***clDice*** (%)↑	***Dice***(%)↑	***IoU***(%)↑	***reError***↓	$\boldsymbol{CD_{\ell_2}}$**(mm)**↓	***reMSE***(1×10^{-4})↓
NeCA	87.01 ± 9.93	90.43 ± 7.46	83.29 ± 11.42	0.139 ± 0.101	0.27 ± 0.37	2.74 ± 2.14
NeCA (90°)	89.07 ± 8.33	**91.03** ± 6.93	**84.17** ± 10.25	**0.111** ± 0.087	**0.22** ± 0.26	**2.73** ± 2.60
3D U-Net	**95.34** ± 4.16	85.18 ± 4.22	74.42 ± 6.24	0.188 ± 0.054	0.31 ± 0.16	4.63 ± 2.91

All values represent mean ± standard deviation.

From the results presented in table 4.3, we can observe that our NeCA model performs better than 3D U-Net model, with relative improvements of 6.16%, 11.92%, 26.06%, 12.90%, and 40.82% in terms of *Dice*, *IoU*, *reError*, CD_{ℓ_2}, and *reMSE* metrics, respectively. 3D U-Net model is better than our NeCA model based on the *clDice* metric, with a respective improvement of 9.57%. 3D reconstruction from two orthogonal projections by our NeCA model produces the best performance in all metrics compared to our NeCA model using two clinical-angle projections. 3D U-Net model maintains the smallest standard deviations among all metrics except for *reMSE*, where our NeCA model performs the best.

Statistical Analysis We present the box plots in figure 4.4 for the evaluation results of all six metrics between our NeCA model and 3D U-Net model at a binarisation threshold of 0.5 on the RCA test dataset. We can see in figure 4.4 that there are more outliers in evaluation of our NeCA model than 3D U-Net. In particular for the evaluation of CD_{ℓ_2}, the outliers in our NeCA model are extremely deviated from the 75th percentile.

With regard to statistical significance test in our work, we use the ASO test [205, 206] to compare score distributions from different models, as described in section 2.4.2. We choose a significance level $\alpha = 0.05$ and $\tau = 0.2$. The confidence scores for all six metrics between our NeCA model and 3D U-Net model using the ASO testing on the RCA test dataset are demonstrated in table 4.4.

From table 4.4, we can find that the score distributions of our NeCA model in terms of *Dice*, *IoU*, *reError*, and *reMSE* are stochastically dominant over the 3D U-Net model. Regarding the metric CD_{ℓ_2}, according to threshold $\tau = 0.2$, our NeCA model is better but

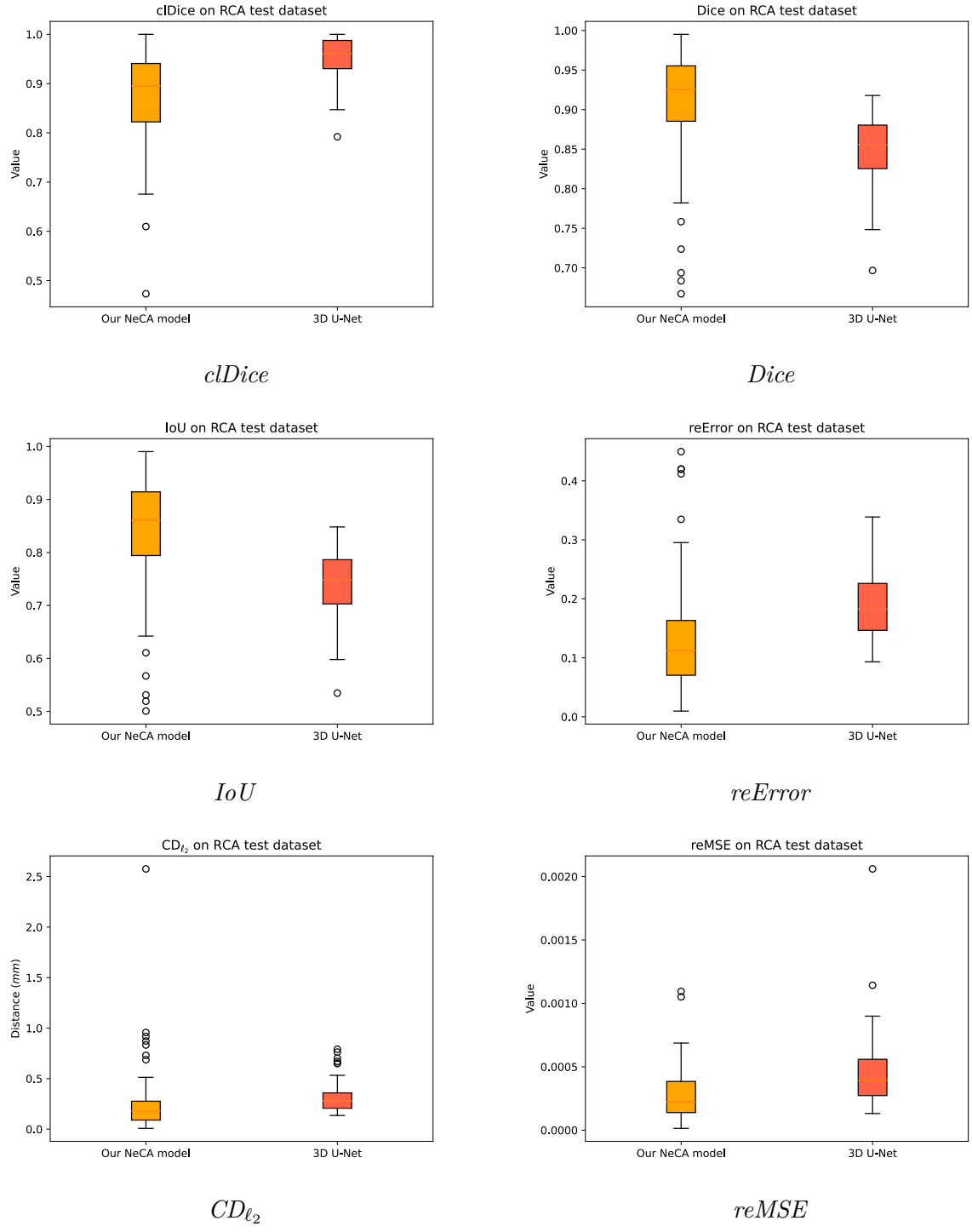


Figure 4.4: Box plots for the evaluation results of all six metrics between our NeCA model and 3D U-Net model at a binarisation threshold of 0.5 on the RCA test dataset.

not stochastically dominant over 3D U-Net. For *clDice*, the 3D U-Net model is found to be stochastically dominant over our NeCA model.

Table 4.4: Confidence scores ϵ_{min} for six metrics between our NeCA model and 3D U-Net model using the ASO testing with a significance level $\alpha = 0.05$ on the RCA test dataset. The confidence scores where our NeCA model is found to be stochastically dominant over 3D U-Net are in **bold**, i.e., $\epsilon_{\text{min}} < \tau = 0.2$.

	clDice	***Dice***	***IoU***	***reError***	$\boldsymbol{CD_{\ell_2}}$	***reMSE***
ϵ_{min}	0.982350	**0.198873**	**0.127973**	**0.0**	0.287172	**0**

Performance Optimisation over Iterations Our NeCA model is optimised for each individual data point. We record the quantitative evaluation results of different metrics every 100 iterations. Here we use three RCA example data points to show how the performance improves iteratively using our NeCA model with both two clinical-angle and orthogonal projections, as illustrated in figure 4.5 and figure 4.6, respectively.

We can see in both figure 4.5 and figure 4.6, the performance does not start to improve until around half of the total iterations, i.e., after $2,500$ iterations. This situation is specially apparent for our NeCA model with orthogonal projections where it only starts to improve after $3,000$ iterations in these three cases. We can also find that usually it takes less than $2,000$ iterations to reach decent results after the start of improvement.

Losses Trend for Supervised 3D U-Net Model We display the trend of training and validation losses with regard to epochs during training for supervised 3D U-Net model on the RCA dataset, as illustrated in figure 4.7. We can find that for a supervised learning model, the performance has a significant improvement just after first several epochs.

Running Time For supervised 3D U-Net model after training, it only takes several seconds to do 3D coronary artery tree reconstruction given input projections, which can be regarded as real-time reconstruction. However, in respect to our NeCA model to optimise for one RCA projection data with reconstruction volume size $128 \times 128 \times 128$ up to $5,000$ iterations, it in average takes 1 hour and 4 minutes.

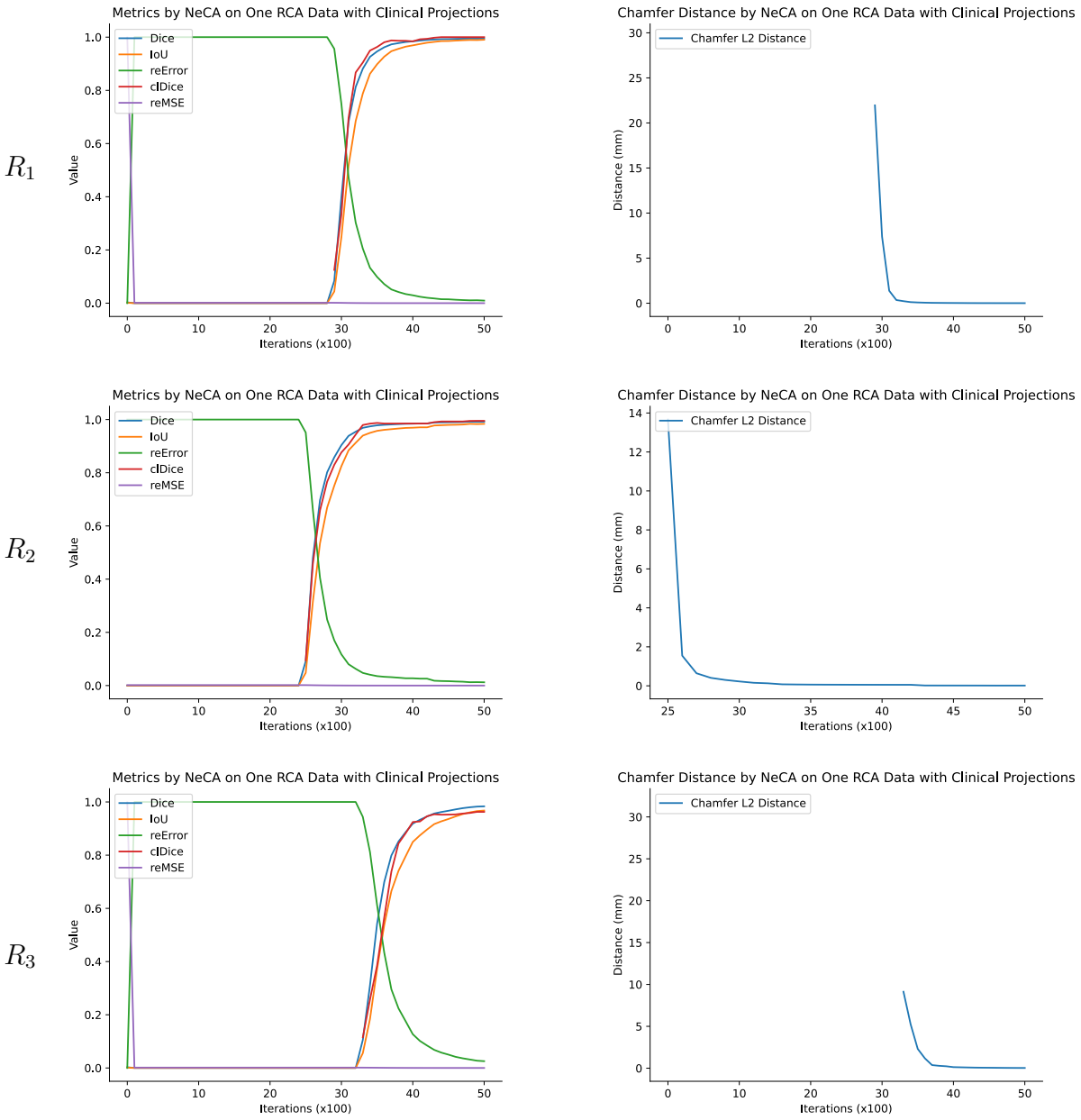


Figure 4.5: Quantitative results of six metrics every 100 iterations for three RCA example data evaluated by our NeCA model with two clinical-angle projections using a binarisation threshold of 0.5. From top to bottom: three RCA data examples $R_{1,2,3}$. From left to right: evaluation results on five metrics (i.e., *clDice*, *Dice*, *IoU*, *reError*, and *reMSE*) and metric CD_{ℓ_2}.

LAD Dataset

Performance over Six Metrics We perform the quantitative evaluations on the LAD test dataset the same as for the RCA data, as described in section 4.3.1. The results are presented in table 4.5.

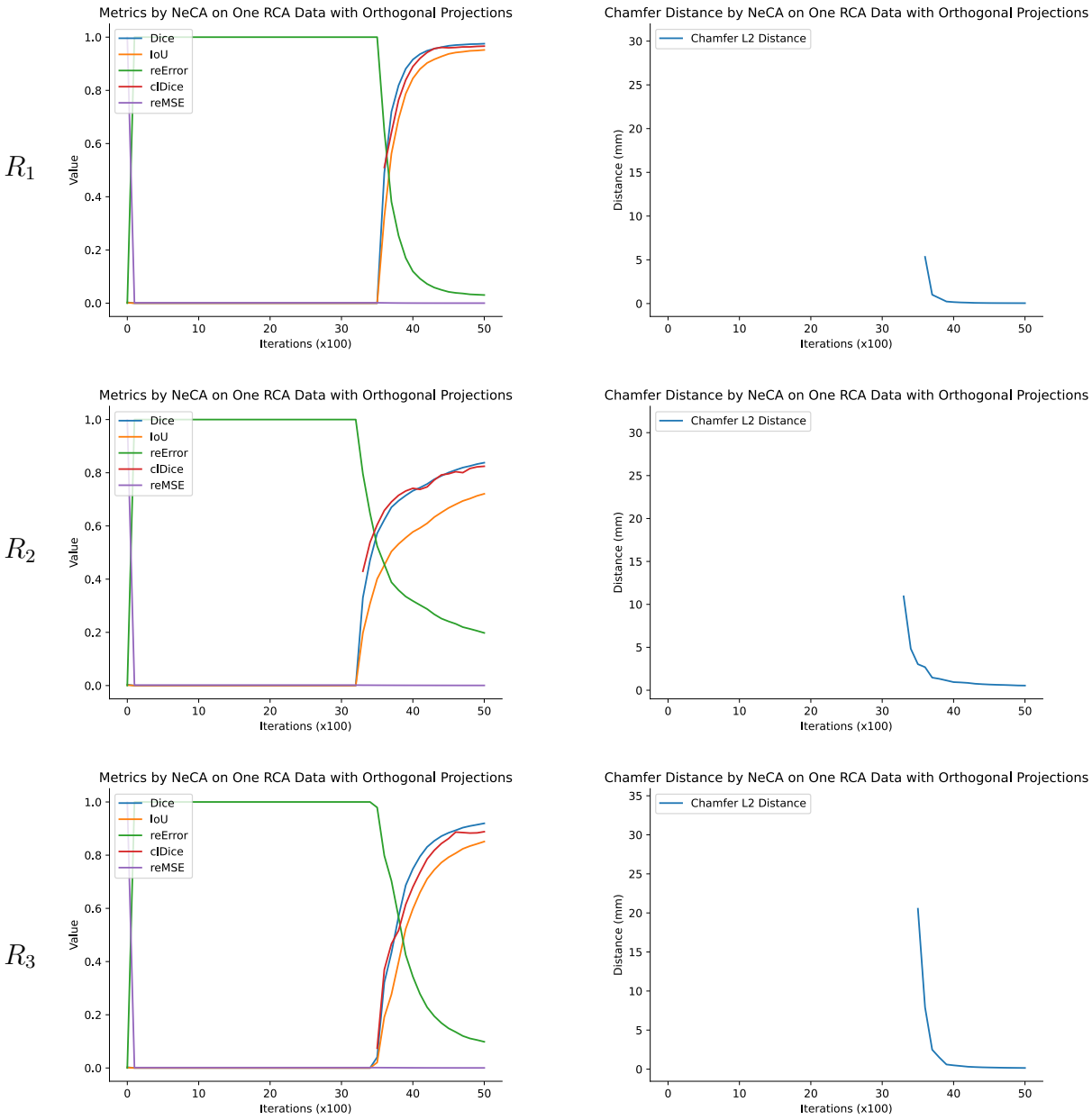


Figure 4.6: Quantitative results of six metrics every 100 iterations for three RCA example data evaluated by our NeCA model with two orthogonal projections using a binarisation threshold of 0.5. From top to bottom: three RCA data examples $R_{1,2,3}$. From left to right: evaluation results on five metrics (i.e., *clDice*, *Dice*, *IoU*, *reError*, and *reMSE*) and metric CD_{ℓ_2}.

In table 4.5, in contrast to the 3D U-Net model, our NeCA model shows improvements of 13.04%, 22.34%, 22.41%, 24.24%, and 29.87% in terms of *Dice*, *IoU*, *reError*, CD_{ℓ_2}, and *reMSE*, respectively. The 3D U-Net model is 9.57% better than our NeCA model with respect to *clDice*. Our NeCA model with two orthogonal projections as input maintains

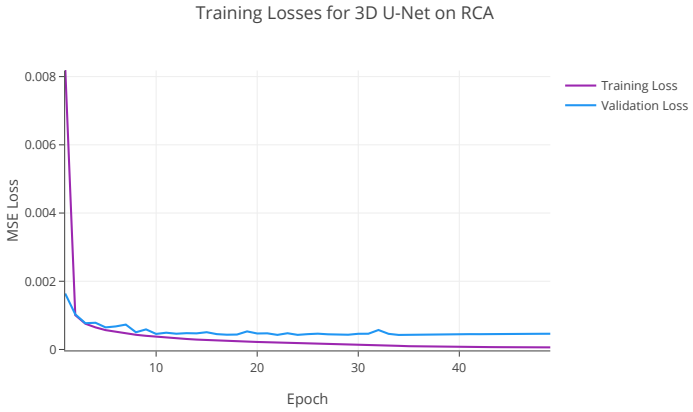


Figure 4.7: Trend of training and validation losses with respect to epochs for supervised 3D U-Net model on the RCA dataset.

Table 4.5: Quantitative evaluation results of NeCA, NeCA (90°), and 3D U-Net model on 79 LAD test data in terms of 6 metrics. Best results of each metric are in **bold**.

Model	***clDice*** **(%)↑**	***Dice*** **(%)↑**	***IoU*** **(%)↑**	***reError*↓**	CD_{ℓ_2}**(mm)↓**	***reMSE*(1 × 10^{-4})↓**
NeCA	76.08 ± 10.42	77.48 ± 9.93	64.28 ± 13.00	0.322 ± 0.129	0.75 ± 0.49	7.28 ± 3.61
NeCA (90°)	**91.69**± 5.62	**94.27** ± 3.91	**89.41**± 6.70	**0.077** ± 0.051	**0.17** ± 0.18	**2.26** ± 1.89
3D U-Net	83.36 ± 7.50	68.54 ± 6.87	52.54 ± 7.91	0.415 ± 0.081	0.99 ± 0.51	10.38 ± 4.22

All values represent mean ± standard deviation.

the best performance among all six metrics compared to both our NeCA model with clinical-angle projections and the 3D U-Net model. Furthermore, our NeCA model with two orthogonal projections as input has the smallest standard deviations among all six metrics compared to both the 3D U-Net model and NeCA with clinical-angle projections.

Statistical Analysis We present the box plots in figure 4.8 of the evaluation results in terms of all six metrics between our NeCA model and 3D U-Net model at a binarisation threshold of 0.5 on the LAD test dataset.

For the statistical significance analysis on the LAD test dataset, we use the ASO test as well, where we choose a significance level of $\alpha = 0.05$ and $\tau = 0.2$. The confidence scores in terms of all six metrics between our NeCA model and the 3D U-Net model are presented in table 4.6.

Table 4.6 demonstrates that our NeCA model evidently outperforms the 3D U-Net model in terms of five metrics, namely *Dice*, *IoU*, *reError*, CD_{ℓ_2}, and *reMSE*. In terms of the

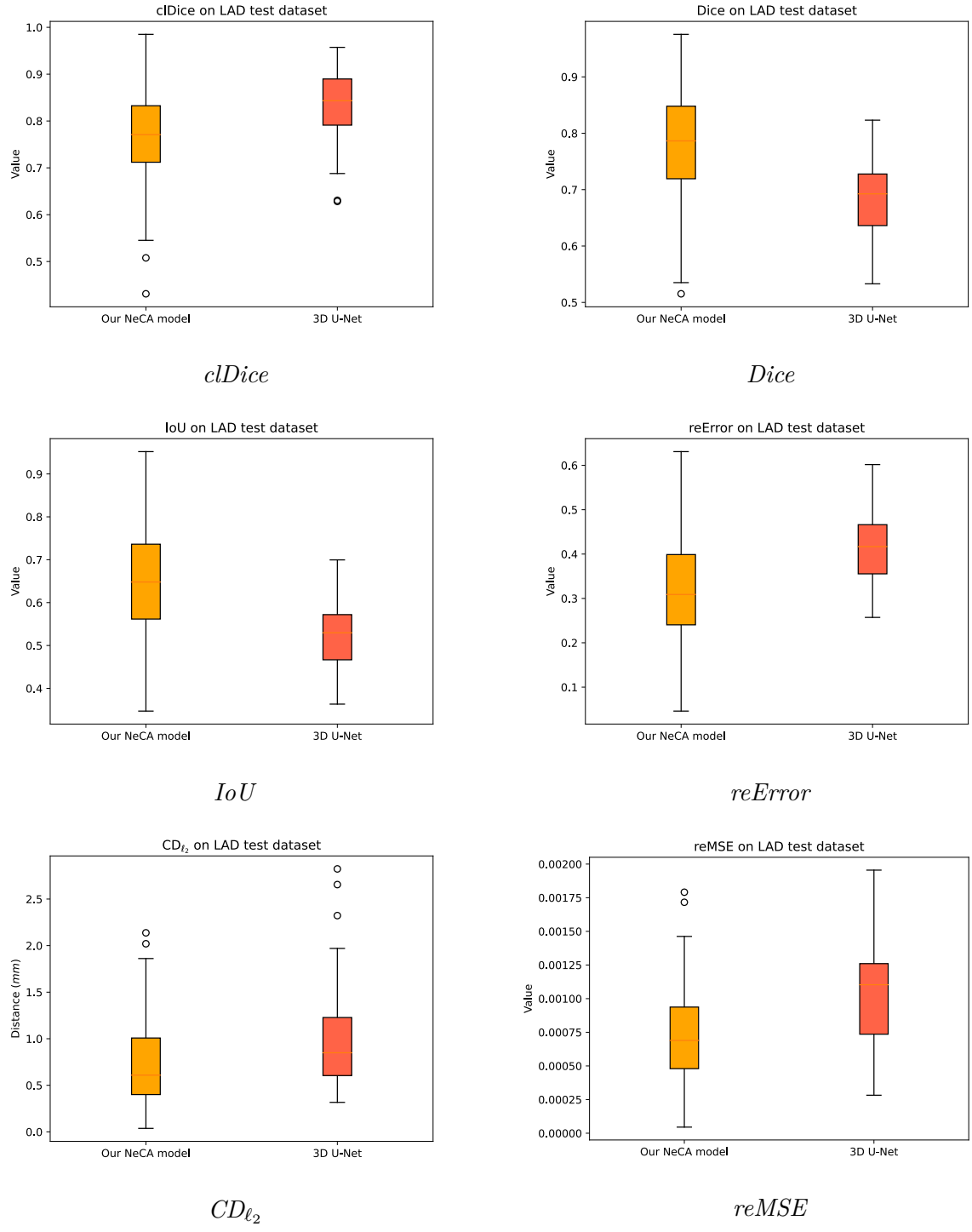


Figure 4.8: Box plots for the evaluation results of all six metrics between our NeCA model and 3D U-Net model at a binarisation threshold of 0.5 on the LAD test dataset.

clDice metric, the 3D U-Net model is stochastically dominant over the NeCA model.

Table 4.6: Confidence scores ϵ_{min} for six metrics between our NeCA model and the 3D U-Net model on the LAD test dataset using ASO testing with a significance level of $\alpha = 0.05$. The confidence scores where our NeCA model is tested to be stochastically dominant over 3D U-Net are in **bold**, i.e., $\epsilon_{\text{min}} < \tau = 0.2$.

	clDice	***Dice***	***IoU***	***reError***	$\boldsymbol{CD_{\ell_2}}$	***reMSE***
ϵ_{min}	0.992092	**0.010340**	**0.005389**	**0**	**0**	**0**

Performance Optimisation over Iterations We record the quantitative evaluation results of six metrics every 100 iterations for each individual data point our NeCA model optimises for. Here we report three LAD example data to demonstrate how the performance enhances iteratively by our NeCA model in terms of both two clinical-angle (as presented in figure 4.9) and orthogonal projections (as presented in figure 4.10) under a binarisation threshold of 0.5.

From figure 4.9 and figure 4.10, we can see that the performance does not start to improve for all three data until at least $2,000$ iterations. It often takes about $2,000$ iterations to reach satisfactory performance after the start of improvement. The same phenomenon is also observed for the RCA dataset in section 4.3.1.

Losses Trend for Supervised 3D U-Net Model We show the trend of training and validation losses in terms of epochs during training for supervised 3D U-Net model on the LAD dataset, as depicted in figure 4.11. We can see the same finding as in section 4.3.1 that it has a large performance improvement just after first several epochs for a supervised learning model.

Running Time The average running time for the 3D coronary artery tree reconstruction ($128 \times 128 \times 128$ volume size) optimisation of each individual LAD projection data by our NeCA model up to $5,000$ iterations is 1 hour and 9 minutes. However, supervised 3D U-Net model can perform almost real-time 3D reconstruction in comparison.

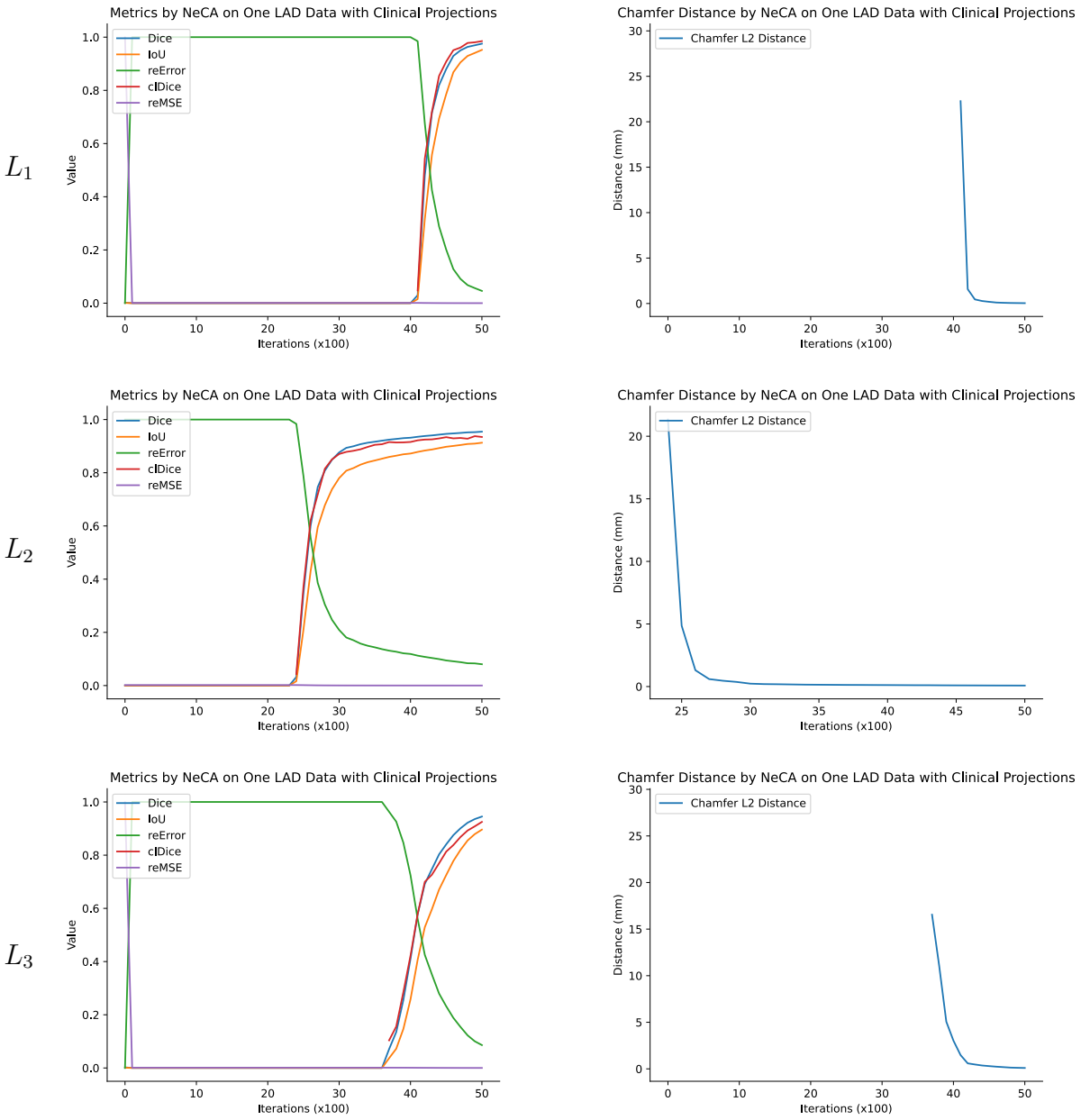


Figure 4.9: Quantitative results of six metrics every 100 iterations for three LAD data evaluated by our NeCA model with two clinical-angle projections under a binarisation threshold of 0.5. From top to bottom: three LAD data examples $L_{1,2,3}$. From left to right: evaluation results on five metrics (i.e., *clDice*, *Dice*, *IoU*, *reError*, and *reMSE*) and metric CD_{ℓ_2}.

4.3.2 Qualitative Results

We present the qualitative results of 3D coronary artery tree reconstruction based on our NeCA model, NeCA (90°), and the 3D U-Net model on both the RCA and LAD test datasets. Here, we use five example data for each dataset. More qualitative results are

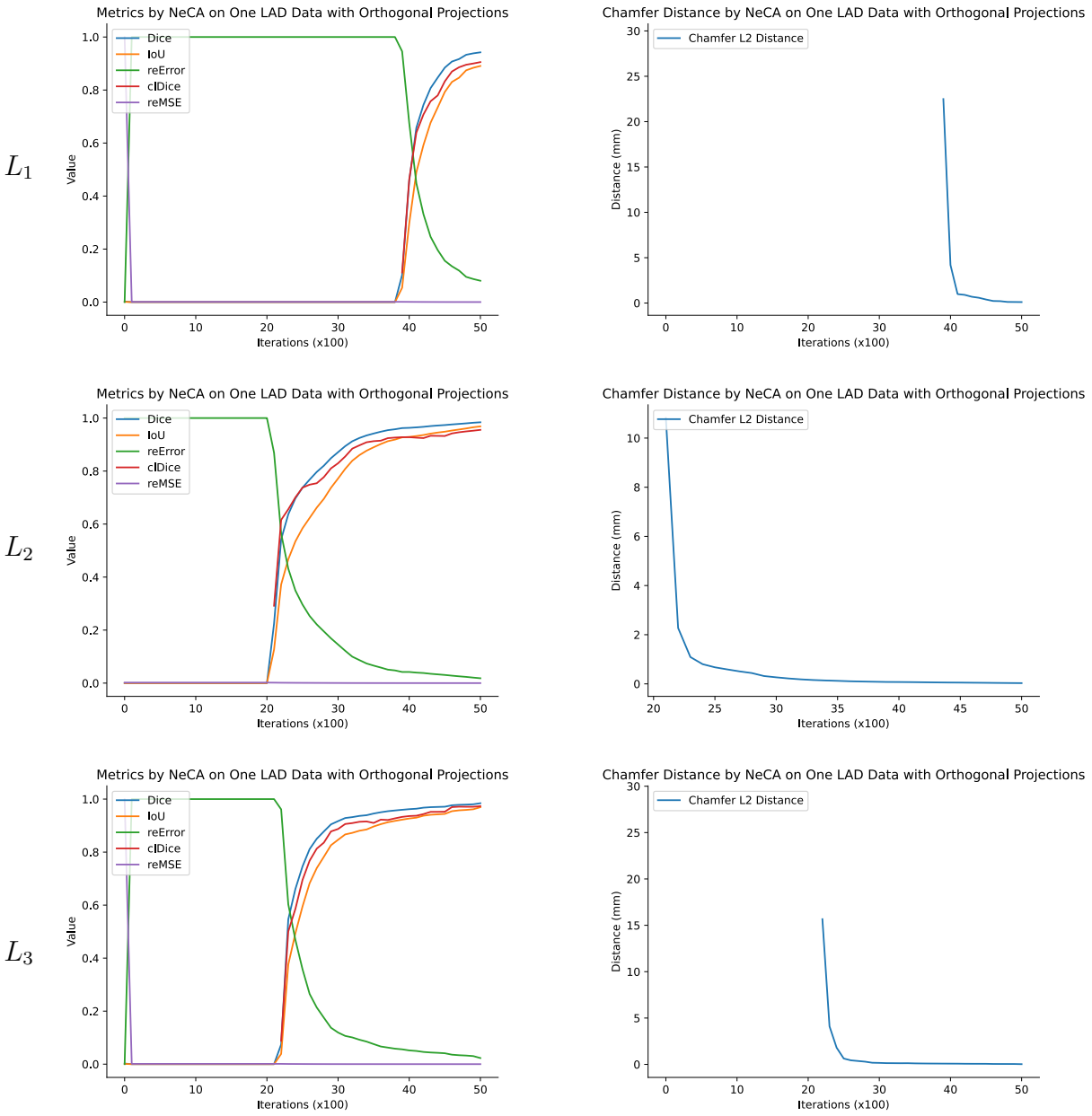


Figure 4.10: Quantitative results of six metrics every 100 iterations for three LAD data evaluated by our NeCA model with two orthogonal projections under a binarisation threshold of 0.5. From top to bottom: three LAD data examples $L_{1,2,3}$. From left to right: evaluation results on five metrics (i.e., *clDice*, *Dice*, *IoU*, *reError*, and *reMSE*) and metric CD_{ℓ_2}.

illustrated in appendix A.

Figure 4.11: Trend of training and validation losses in terms of epochs for supervised 3D U-Net model on the LAD dataset.

RCA Dataset

3D Reconstruction Results Figure 4.12 illustrates five RCA examples of 3D coronary artery tree reconstruction using our NeCA model, NeCA (90°), and 3D U-Net model, along with the corresponding ground truth for each case. The results show that all three models can successfully perform satisfactory 3D RCA reconstruction.

Comparison Between 3D Reconstruction and Ground Truth We additionally compare the five 3D RCA reconstruction results using the NeCA, NeCA (90°), and 3D U-Net model with the corresponding ground truth in the same 3D space, as illustrated in figure 4.13. These figures show that our NeCA model demonstrates better reconstruction overlap than the 3D U-Net model.

LAD Dataset

3D Reconstruction Results We show in figure 4.14 five 3D LAD reconstruction results using our NeCA model, NeCA (90°), and the 3D U-Net model, with the corresponding ground truth. From the results, we can observe that our NeCA model successfully reconstructs the LAD vasculature in all the cases. On the other hand, the 3D U-Net model fails to reconstruct some branches in $L_{3,4,5}$ and loses vessel connectivity, as presented in red boxes.

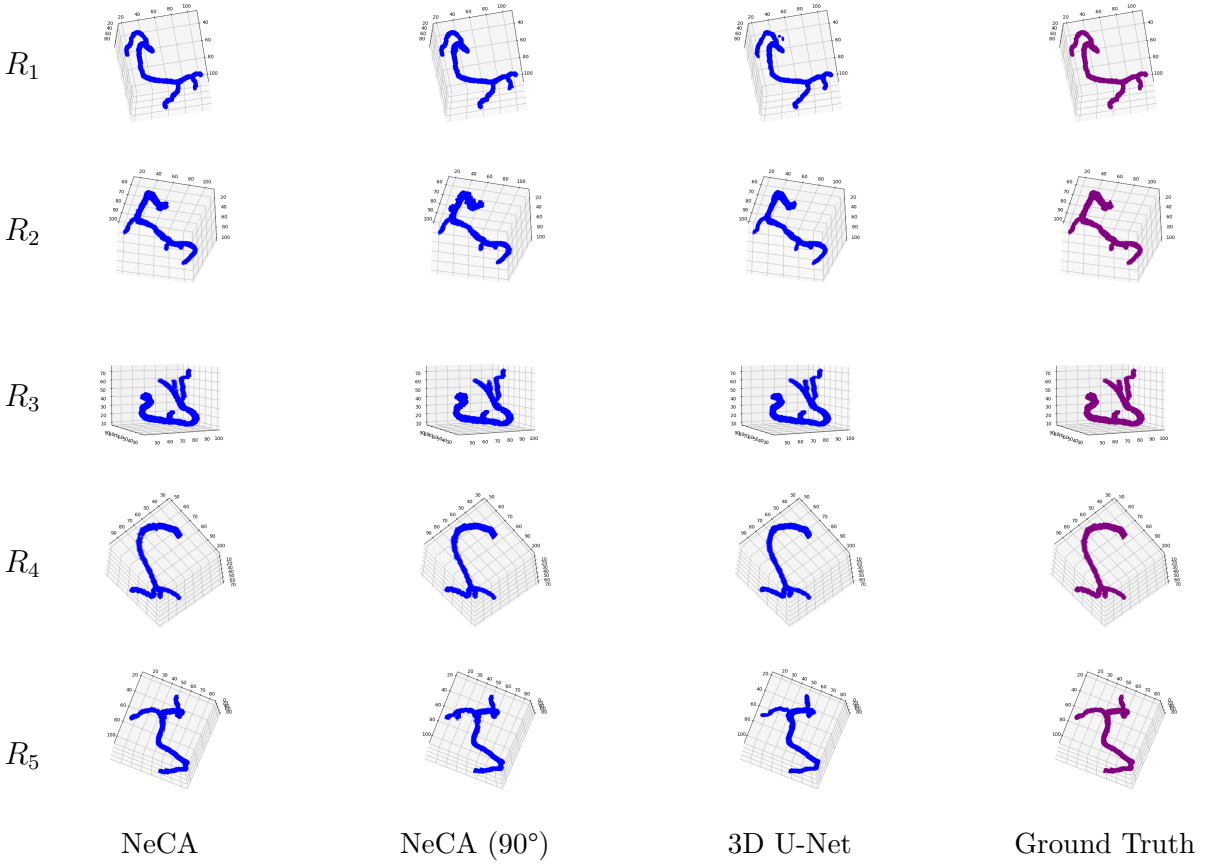


Figure 4.12: Five qualitative results of 3D RCA reconstruction with a binarisation threshold of 0.5. From top to bottom: five RCA data points $R_{1,2,3,4,5}$. From left to right: reconstruction results from our NeCA model, our NeCA model using two orthogonal projections (90°), and 3D U-Net model, along with the corresponding ground truth.

Comparison Between 3D Reconstruction and Ground Truth We also compare in figure 4.15 the five 3D LAD reconstruction results using NeCA, NeCA (90°), and the 3D U-Net models with the corresponding ground truth in the same 3D space. The results show similar performance to the RCA dataset; our NeCA model demonstrates better reconstruction overlap than the 3D U-Net model.

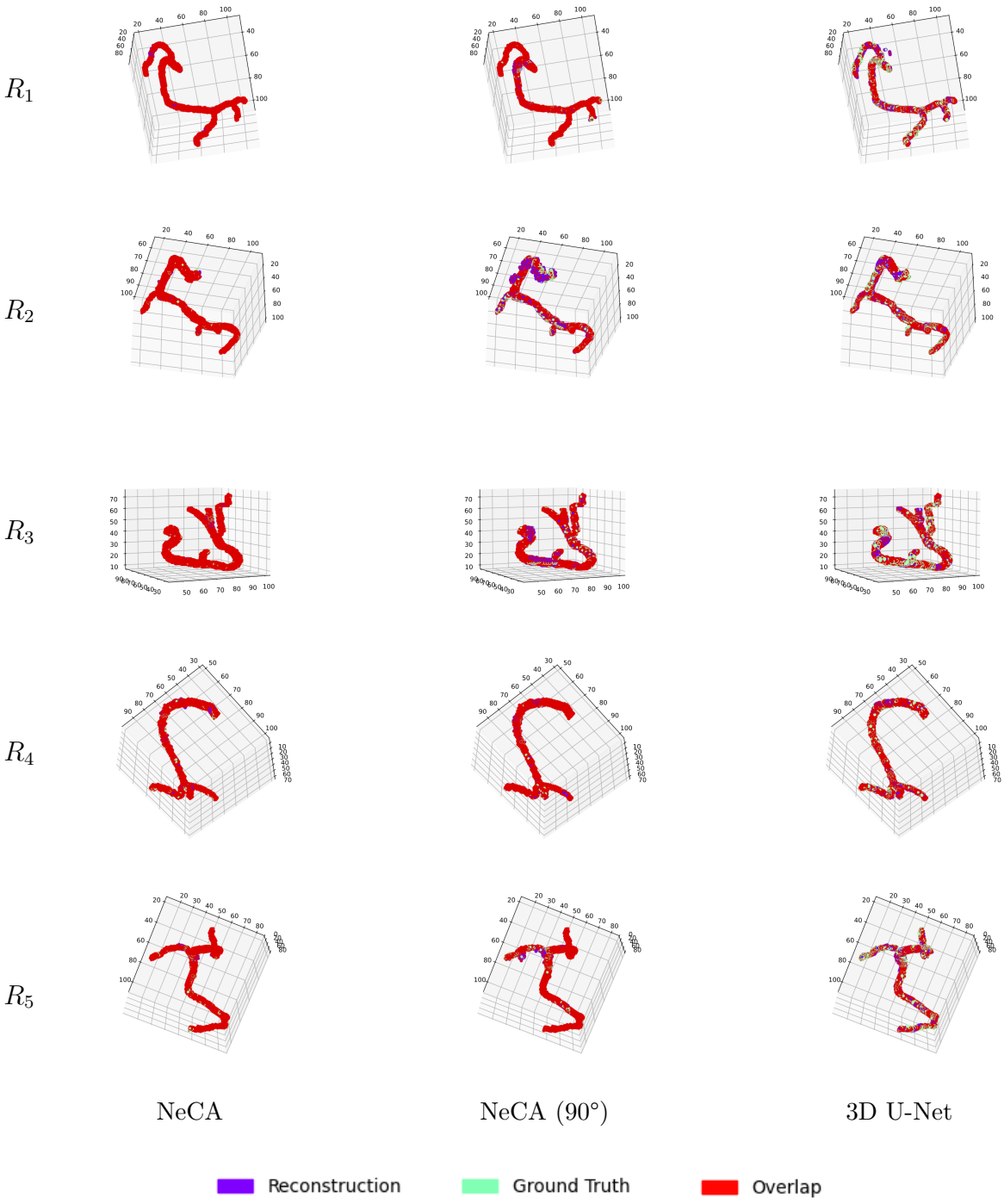


Figure 4.13: Five 3D RCA reconstruction results by a binarisation threshold of 0.5 compared with the corresponding ground truth in the same 3D space. From top to bottom: five RCA data $R_{1,2,3,4,5}$. From left to right: comparison results from our NeCA model, our NeCA model using two orthogonal projections (90°), and 3D U-Net model. Colour purple represents the reconstruction results, colour green is the ground truth, and colour red means the overlap between them.

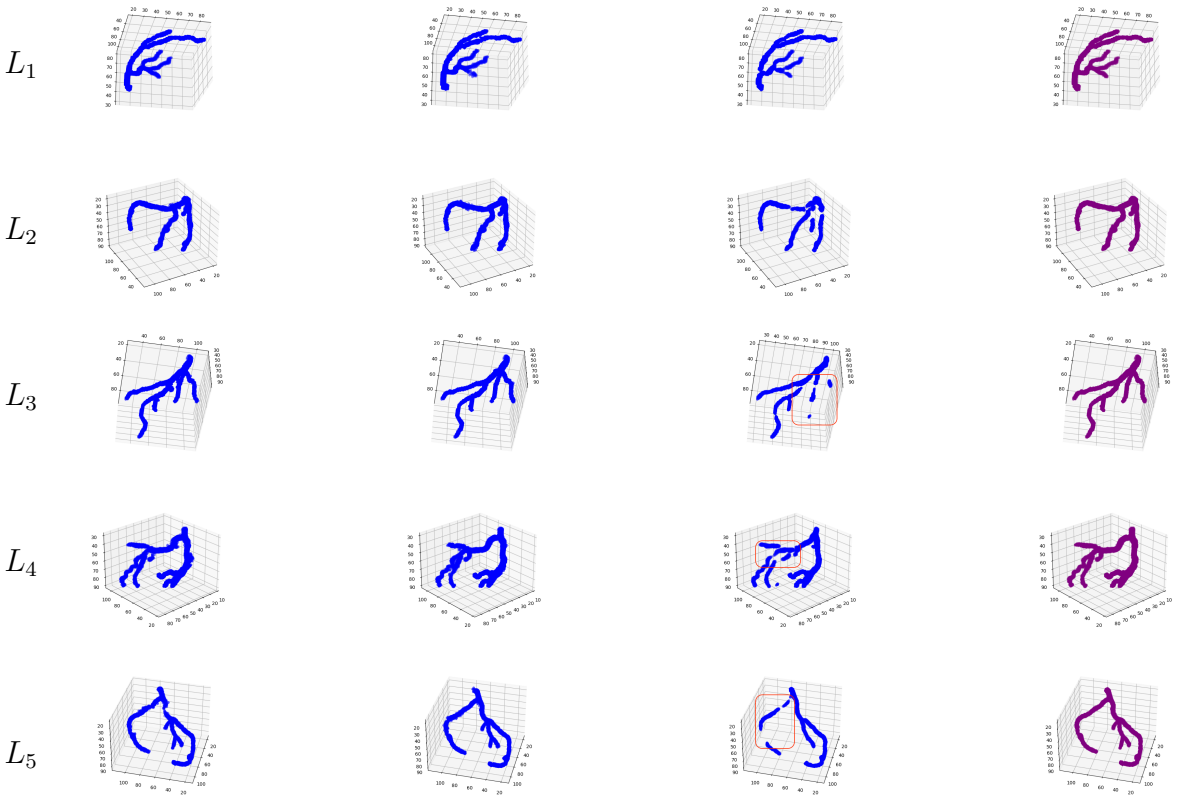


Figure 4.14: Five qualitative 3D LAD reconstruction results with a binarisation threshold of 0.5. From top to bottom: Five LAD data $L_{1,2,3,4,5}$. From left to right: reconstruction results from our NeCA model, our NeCA model using two orthogonal projections (90°), and 3D U-Net model, along with the corresponding ground truth.

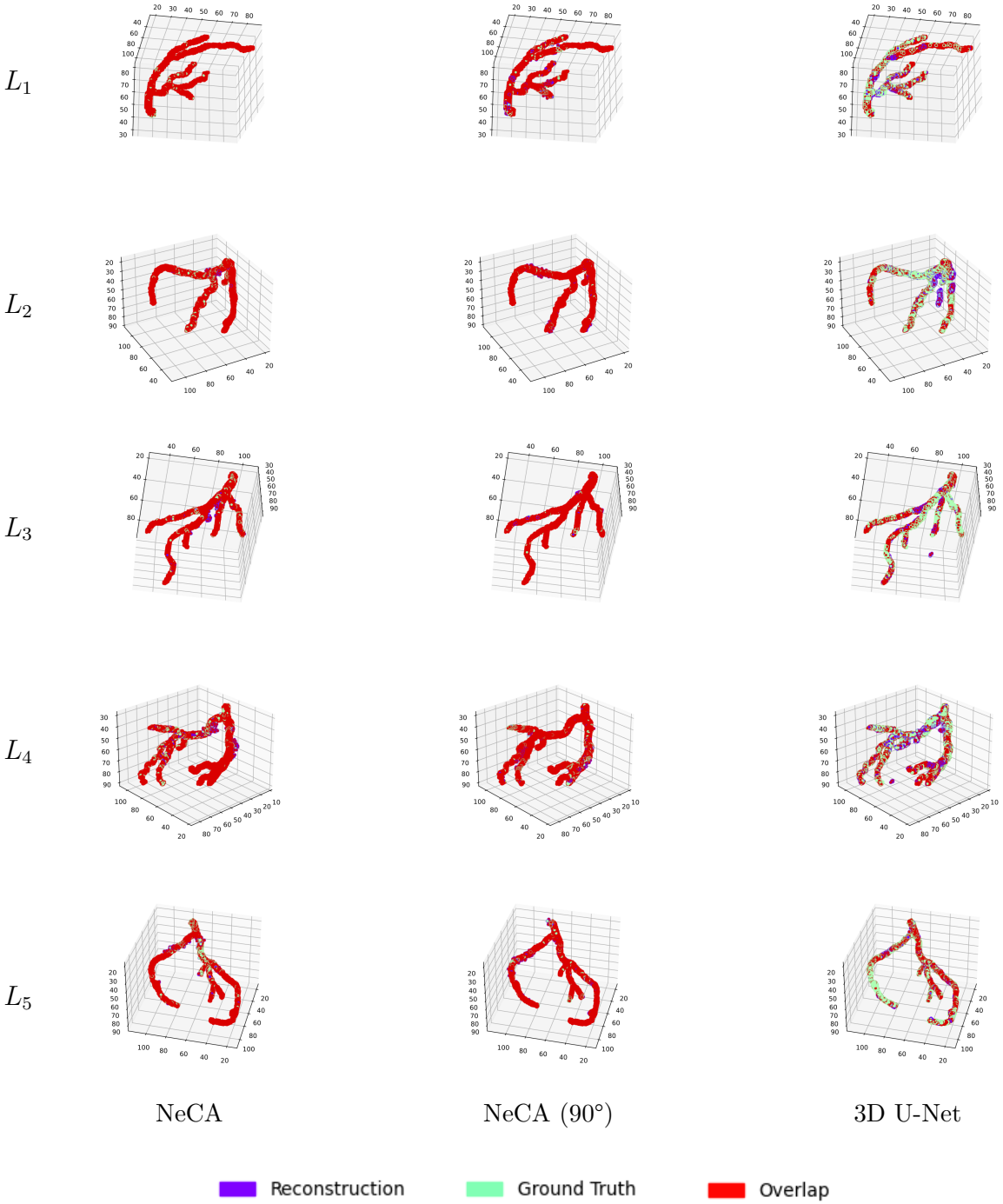


Figure 4.15: Five 3D LAD reconstruction results by a binarisation threshold of 0.5 compared with the corresponding ground truth in the same 3D space. From top to bottom: five LAD data $L_{1,2,3,4,5}$. From left to right: comparison results from our NeCA model, our NeCA model using two orthogonal projections (90°), and 3D U-Net model. Colour purple represents the reconstruction results, colour green is the ground truth, and colour red means the overlap between them.

4.4 Discussion and Conclusion

Our evaluation on both the RCA and LAD datasets demonstrates that the NeCA model performs better than the supervised 3D U-Net model in terms of five metrics: *Dice*, *IoU*, *reError*, CD_{ℓ_2}, and *reMSE*. The NeCA model performs statistically significantly better than 3D U-Net model in four metrics for the RCA dataset and five metrics for the LAD dataset out of a total of six metrics. This indicates that our self-supervised learning model, where neither 3D ground truth for supervision nor large training datasets are required, is better than the supervised 3D U-Net model in 3D coronary tree reconstruction from only two projections. It is also demonstrated qualitatively in section 4.3.2 that our NeCA model presents good vasculature reconstruction. In addition, due to the intrinsic properties of our model, we do not need to train two models for RCA and LAD separately, and as a result, it has significant potential to generalise to other tasks.

Our model optimised with two orthogonal projections (NeCA (90°)) shows consistently better performance than our model with two clinical-angle projections (table 4.5), since two orthogonal projections usually contain more feature coverage and less overlapped redundant information (figure 4.1). However, in real clinics such as cardiac catheterisation laboratories, projections are generally not acquired at orthogonal views, thus necessitating this feature of our NeCA model.

Our NeCA model contains two trainable components: the hash tables with feature vectors $\boldsymbol{\Theta}$ from the multiresoultion hash encoder and network parameters $\boldsymbol{\Phi}$ from the residual MLP. The residual MLP is the backbone of the neural implicit representation, so it cannot be replaced. For the multiresolution hash encoder to encode the coordinates, there are alternative encoders available, such as a frequency encoder, which is not learnable. We have tested the coordinate encoder where we have replaced our multiresolution encoder with a frequency encoder and used the same projection geometry for validation. According to our experiments, the model could not reconstruct any vessels for every case of the RCA and LAD datasets under $5,000$ iterations.

The supervised 3D U-Net model, once trained, can perform real-time 3D coronary tree reconstruction, while our model takes around one hour to optimise the results with a volume size of $128 \times 128 \times 128$ for $5,000$ iterations. We have also tested our model to optimise a coronary tree of size $64 \times 64 \times 64$, which takes on average of 11 minutes for reconstruction. Therefore, there is a tradeoff between lower reconstruction time and better reconstruction resolution for our NeCA model. The 3D U-Net model applies a pretrained model during evaluation, so when reconstructing out-of-distribution data, it may fail to generalise, which is a serious threat during clinical applications, whereas our model is optimised for each individual data points and can generalise well. Hence, there is also a tradeoff between real-time reconstruction and stable performance between the 3D U-Net and our NeCA model.

The input cone-beam projections to our NeCA model are based on simulation of X-ray intensity attenuation though the object, i.e., the 3D coronary artery tree. In our experiments using 3D segmented CCTA data, the attenuation coefficients for the coronary tree are assumed to be uniform as a value of 1. However, in real scenarios, the actual coefficients vary, usually within a certain range due to different vessel conditions. Moreover, blood and contrast injected in the vessel contribute to the X-ray attenuation as well as the other tissues and organs in the background. Though the background removal could be solved with automated coronary vessels segmentation [238, 239], the 3D coronary artery tree reconstruction based on real X-ray projections with contrast injected and different vessel conditions needs to be explored further. In addition, during cardiac interventions based on single-plane X-ray angiography systems, cardiac and respiratory motions exist between different projections, which are another critical factors affecting the reconstruction process.

In summary, we have proposed a self-supervised deep learning method, NeCA, using neural implicit representation to achieve 3D coronary artery tree reconstruction from only two projections. Our method neither requires 3D ground truth for supervision nor large training datasets and optimises the reconstruction results in an iterative self-supervised

fashion with only the projection data of one patient as input. We leverage the advantages of a learnable multiresolution hash encoder [99] to allow for efficient feature encoding, residual MLP neural networks as a continuous function to represent the coronary artery tree in 3D space, and a differentiable projector layer [221] to enable self-supervised learning from 2D input projections. We use a public CCTA dataset [222] containing both RCA and LAD data to validate our model's feasibility on the task based on six quantitative metrics, and we perform a thorough evaluation. The results demonstrate that our proposed NeCA model achieves promising performance in both vessel topology preservation and branch-connectivity maintenance compared to the supervised 3D U-Net model. Our proposed model also has a high possibility to generalise to other clinical tasks where the ground truth is usually unavailable and hard to acquire.

Chapter 5

Reconstruction on Real-world Data

Abstract - DeepCA: Deep Learning-based 3D Coronary Artery Tree Reconstruction from Two 2D Non-simultaneous X-ray Angiographic Projections

SOTA approaches for 3D coronary artery tree reconstruction from a few projections require significant manual interactions and cannot correct the non-rigid cardiac and respiratory motions between non-simultaneous projections. In this chapter, a novel deep learning pipeline named DeepCA is proposed. DeepCA leverages the Wasserstein conditional generative adversarial network with gradient penalty, latent convolutional transformer layers, and a dynamic snake convolutional critic to implicitly compensate for the non-rigid motion and provide 3D coronary artery tree reconstruction. Through simulating projections from CCTA data, the generalisation of 3D coronary artery tree reconstruction on real non-simultaneous ICA projections is achieved, in contrast to NeCA in chapter 4 which assumes no motion exists between projections. An application-specific evaluation metric is incorporated to validate the proposed DeepCA model on both a CCTA dataset and a real ICA dataset, together with Chamfer ℓ_2 distance. The results demonstrate the promising performance of the DeepCA model in vessel topology preservation, recovery of missing features, and generalisation ability to real ICA data. To the best of our knowledge, this is the first study that leverages deep learning to achieve 3D coronary artery tree reconstruction from two real non-simultaneous X-ray angiographic projections acquired from single-plane X-ray angiography systems. The implementation of this work is available at: `https://github.com/WangStephen/DeepCA`.

Publication: Wang, Y., Banerjee, A., Choudhury, R., & Grau, V., "DeepCA: Deep Learning-based 3D Coronary Artery Tree Reconstruction from Two 2D Non-simultaneous X-ray Angiography Projections," IEEE/CVF Winter Conference on Applications of Computer

Vision (WACV) 2025. (**Oral**)

5.1 Introduction

3D coronary artery tree reconstruction from real ICA images poses significant challenges: the complex vascular shape and limited projections provide limited information on 3D vessel structures. Most importantly, due to the non-simultaneous image acquisition, significant cardiac and respiratory non-rigid motions cause vessels to misalign between projections, which aggravates the difficulties of 3D reconstruction. Biplane X-ray angiography systems can simultaneously capture two projections and hence, unaffected by such non-rigid motions; however, they are expensive for clinical usage. Many traditional methods have been proposed for 3D coronary artery tree reconstruction from 2D non-simultaneous X-ray projections [4]. However, most of these methods are dependent on traditional stereo-vision algorithms, which usually require significant manual interactions such as label annotations, and they often cannot correct non-rigid cardiac motion. In terms of 3D coronary artery tree reconstruction using deep learning, previous methods have typically used synthetic data, CCTA data, or ICA data from bi-planar scans, none of which suffer from non-rigid motion between projections [168–173, 240]. This limitation makes previous methods ill-suited for real non-simultaneous ICA acquisitions. Despite the improvement in deep neural networks, 3D coronary artery tree reconstruction from limited non-simultaneous angiographic projections has remained an open problem.

In this chapter, we propose a novel deep learning pipeline named DeepCA, leveraging the Wasserstein conditional generative adversarial network with gradient penalty, latent convolutional transformer layers, and a dynamic snake convolutional critic to implicitly compensate for the non-rigid motion to achieve 3D coronary artery tree reconstruction from two real non-simultaneous ICA projections. To resemble real non-simultaneous ICA projections, we simulate 2D projections in different planes from CCTA data containing real coronary tree geometries, with a rigid transformation applied to the CCTA data before forward projection on the second projection plane. We then use these simulated projections

to learn from the CCTA ground truth to enable generalisation to real non-simultaneous ICA projections. In this way, we overcome the problems of both the limited number of real paired ICA data with projection geometry information and the unavailable 3D ground truth for real ICA data. We focus on the RCA in this study, because RCA undergoes more compressive strain and is affected more by motion artifacts than other coronary vessels. We provide an application-specific evaluation method to tackle the deformation in 3D reconstructions, unavailability of 3D ground truth for real ICA scans, and motion between projection planes, together with Chamfer ℓ_2 distance. We validate our proposed model on a CCTA dataset and a real ICA dataset (unseen domain), in comparison to four other models. The evaluation results demonstrate the promising performance of our proposed model in vessel topology preservation, recovery of missing features, and generalisation ability in 3D coronary tree reconstruction from real non-simultaneous ICA projections. The main contributions of this work are:

1. **3D coronary tree reconstruction using deep learning:** To the best of our knowledge, this is the first study that leverages deep learning to achieve 3D coronary artery tree reconstruction from two real non-simultaneous angiographic projections.
2. **Generalisation:** Through simulating projections from CCTA data, we achieve generalisation on 3D coronary artery tree reconstruction from two real non-simultaneous ICA projections.
3. **Extensive evaluation:** We use a specific metric designated for this problem in the absence of motion-free 3D ground truth, which provides a baseline for future improvement in this area.

5.2 Proposed Pipeline

Our proposed DeepCA method consists of two blocks: a data preprocessing block and a 3D reconstruction with motion compensation block, as illustrated in figure 5.1. In the data preprocessing block, we generate two simulated ICA projections based on 3D CCTA data,

with simulated motion on the second projection plane, and then apply backprojection on them to produce the input to the model at the next block. In the 3D reconstruction with motion compensation block, we map the 3D backprojection input to the CCTA data for 3D coronary artery tree reconstruction via training a deep neural network, implicitly compensating for any motion.

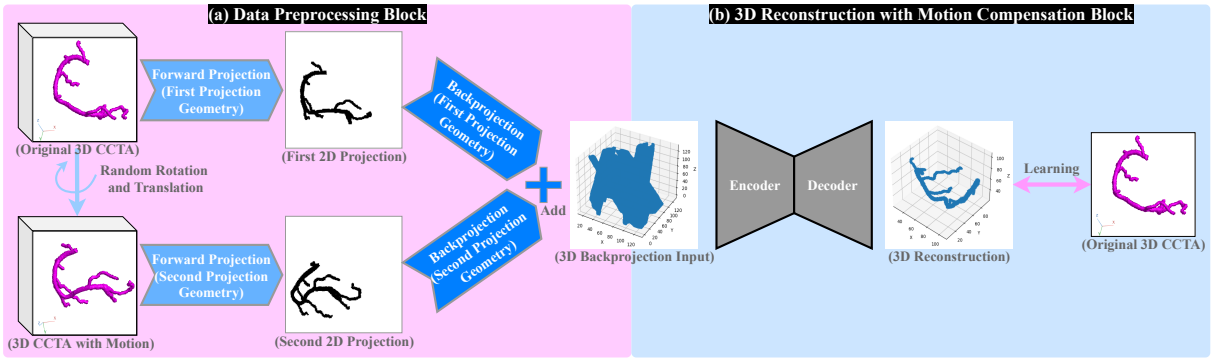


Figure 5.1: Overall workflow of our proposed DeepCA pipeline consists of a data preprocessing block and a 3D reconstruction with motion compensation block. **(a)** The data preprocessing block generates two simulated ICA projections from 3D CCTA data, including simulated motion between projections, and then produces the 3D model input via performing backprojection on the two simulated projections. **(b)** The 3D reconstruction with motion compensation block receives the 3D backprojection input to train a deep neural network for 3D coronary artery tree reconstruction learned from the CCTA data, implicitly compensating for any motion.

5.2.1 Data Preprocessing Block

Breathing and cardiac motions introduce deformations to the coronary tree between projections. We use deep learning to implicitly compensate for these motion artifacts. In the generation of simulated projections, we introduce rigid transformations to the CCTA data before performing forward projection on the second projection plane to simulate motion, as illustrated in figure 5.1. The CCTA data with motion is used on the second projection plane to simulate the breathing and cardiac motions; here we rotate the CCTA data randomly for both primary and secondary angles ranging from -10° to 10° and add translations of -8 mm to 8 mm in both horizontal and vertical directions. We use the projection geometry of real coronary angiography to simulate the two cone-beam forward projections. Details of the projection geometry parameters are provided in table 5.1.

Table 5.1: Projection geometry to simulate cone-beam forward projections on the CCTA dataset, in order to resemble the real ICA settings. The CCTA data with motion is used on the second projection plane to simulate the breathing and cardiac motions; here we rotate the CCTA data randomly for both primary and secondary angles ranging from -10° to 10° and add translations of -8 mm to 8 mm in both horizontal and vertical directions.

	First Projection Plane	Second Projection Plane
Phantom	Original 3D CCTA Data	3D CCTA Data with Motion
Detector Spacing	$0.2769 \times 0.2769\ mm^2$ to $0.2789 \times 0.2789\ mm^2$	
Detector Size	512×512	
Volume Spacing	$90 \times 90 \times 90\ mm^3$ to $105 \times 105 \times 105\ mm^3$	
Volume Size	$128 \times 128 \times 128$	
DSD	970 mm to 1010 mm	1050 mm to 1070 mm
DSO	745 mm to 785 mm	$\pm$ 3 mm to the First Projection
Primary Angle	18° to 42°	−8° to 8°
Secondary Angle	−8° to 8°	18° to 42°

Since contrast injections used in real ICA projections change image intensity values between projections, we first segment vessels from the images to binarise them, where points on vessels are assigned as 1 and background as 0. In order to generalise to real ICA projections, we binarise the simulated projections with a threshold of 0 as well, i.e. any points with values greater than 0 are set to 1 and 0 otherwise. Using the known projection geometry, we perform backprojection on both binary simulated projections separately. We binarise the two 3D backprojections with a threshold of 0 and add them together to generate a single 3D input to our model.

5.2.2 3D Reconstruction with Motion Compensation Block

We train a model to map the 3D backprojection result to its corresponding CCTA ground truth. Our DeepCA model architecture is based on the Wasserstein conditional generative adversarial network (WCGAN) with gradient penalty, latent convolutional transformer layers, and a dynamic snake convolutional critic, as illustrated in figure 5.2. Via mapping the input with non-aligned projections to 3D coronary tree data, most motion artifacts are corrected by our model. With the critic used, any residual uncorrected deformations are adjusted, while ensuring the connectedness of the coronary tree structures in the

reconstructions and increasing the model's elastic generalisation capacity. So when generalising to real ICA projections, the non-rigid motion is compensated implicitly.

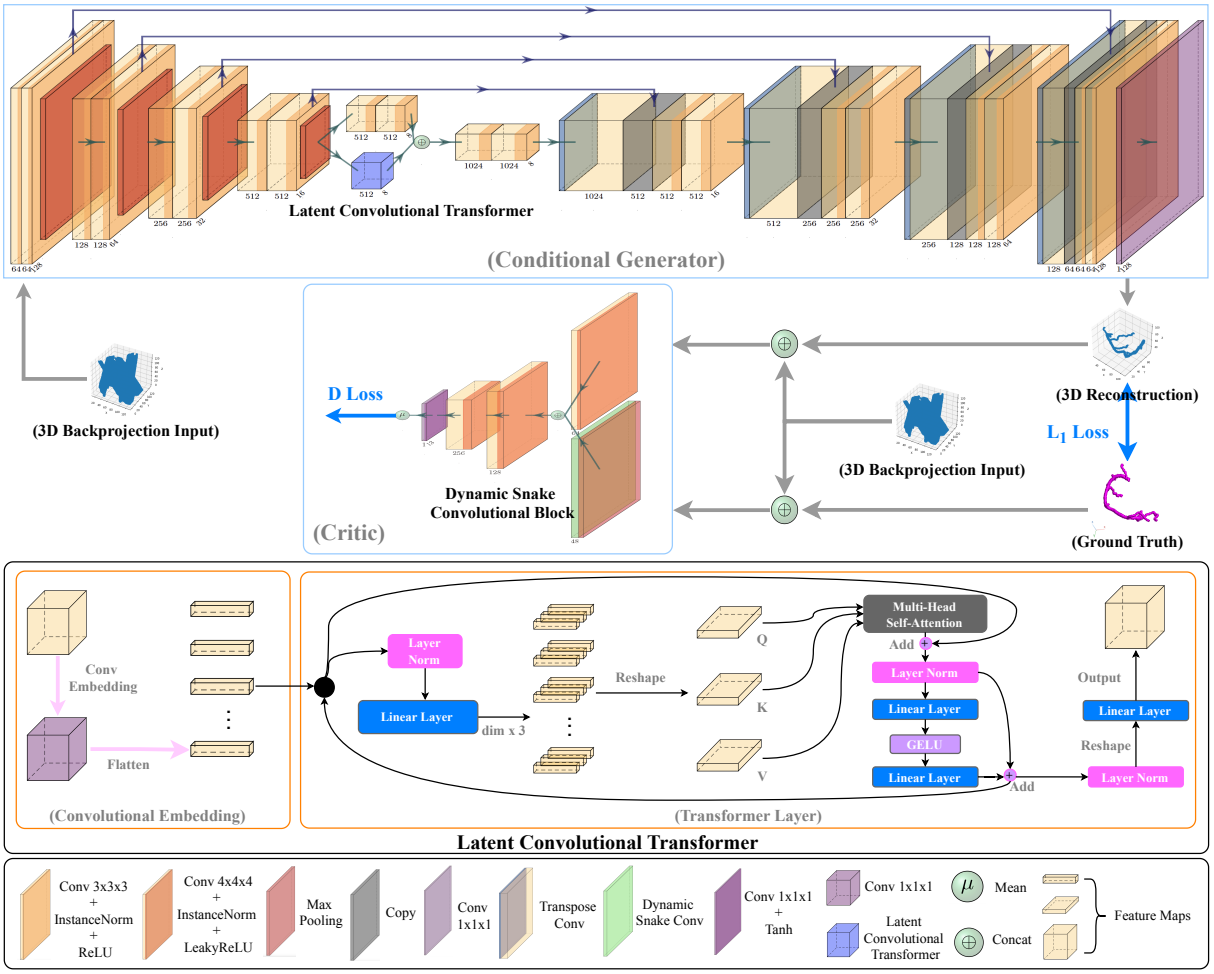


Figure 5.2: Proposed DeepCA model architecture includes a conditional generator and a critic. The conditional generator is based on 3D U-Net with additional proposed convolutional transformer layers in the latent space. The generator produces corresponding reconstructed results according to the input condition. The latent convolutional transformers are built on convolutional embeddings following 8 transformer layers. The predicted results and the corresponding ground truth are concatenated with the input separately, which are then sent to the critic. The proposed critic uses both dynamic snake convolution and traditional convolution at the first layer to extract both global tubular and local features, and then applies several downsamplings to generate the critic loss.

WCGAN with Gradient Penalty (WCGAN-GP)

The conditional structure of the GAN [60] enables us to generate a desired output from a specific input, and the Wasserstein adversarial objective with an additional gradient penalty constraint [59] improves training stability. The model consists of an encoder-

decoder generator G and a critic D. The 3D backprojection result $\mathbf{x}$ is the input to the generator G, which has the 3D U-Net [237] as backbone, producing the predicted reconstruction $\widehat{\mathbf{y}}$ as output. To ensure strict learning from the 3D backprojection input $\mathbf{x}$ to the corresponding ground truth $\mathbf{y}$, we keep the input $\mathbf{x}$ without any added noise, in contrast to previous style transfer applications. The predicted reconstruction $\widehat{\mathbf{y}}$ and the corresponding ground truth $\mathbf{y}$ are then concatenated ($\oplus$) with the conditional input $\mathbf{x}$, respectively.

$$\widehat{\mathbf{y}} = G(\mathbf{x}), \quad \widehat{\mathbf{y}}_\mathbf{x} = \widehat{\mathbf{y}} \oplus \mathbf{x}, \quad \mathbf{y}_\mathbf{x} = \mathbf{y} \oplus \mathbf{x}. \tag{5.1}$$

Next, $\widehat{\mathbf{y}}_\mathbf{x}$ and $\mathbf{y}_\mathbf{x}$ are used in the critic to approximate the Wasserstein distance (or, Earth-Mover distance) $W((\mathbb{P}_r)_{\mathbf{y}_\mathbf{x}\sim\mathbb{P}_r}, (\mathbb{P}_g)_{\widehat{\mathbf{y}}_\mathbf{x}\sim\mathbb{P}_g})$ and gradient penalty constraint $GP(\mathbb{P}_{\mathbf{y}_\mathbf{x}})$ for each data batch, where $\mathbb{P}_r$ is the conditional ground truth data distribution, $\mathbb{P}_g$ is the conditional model generation distribution, and $\mathbb{P}_{\mathbf{y}_\mathbf{x}}$ is the distribution sampling uniformly along straight lines between pairs of points sampled from the distributions $\mathbb{P}_r$ and $\mathbb{P}_g$.

$$W((\mathbb{P}_r)_{\mathbf{y}_\mathbf{x}\sim\mathbb{P}_r}, (\mathbb{P}_g)_{\widehat{\mathbf{y}}_\mathbf{x}\sim\mathbb{P}_g}) = \mathop{\mathbb{E}}_{\mathbf{y}_\mathbf{x}\sim\mathbb{P}_r}[D(\mathbf{y}_\mathbf{x})] - \mathop{\mathbb{E}}_{\widehat{\mathbf{y}}_\mathbf{x}\sim\mathbb{P}_g}[D(\widehat{\mathbf{y}}_\mathbf{x})]. \tag{5.2}$$

$$GP(\mathbb{P}_{\mathbf{y}_\mathbf{x}}) = \mathop{\mathbb{E}}_{\mathbf{y}_\mathbf{x}\sim\mathbb{P}_{\mathbf{y}_\mathbf{x}}}[(\|\nabla_{\mathbf{y}_\mathbf{x}} D(\mathbf{y}_\mathbf{x})\|_2 - 1)^2], \text{where } \mathbf{y}_\mathbf{x} = \epsilon\mathbf{y}_\mathbf{x} + (1-\epsilon)\widehat{\mathbf{y}}_\mathbf{x}, \quad \epsilon \in U[0,1]. \tag{5.3}$$

During training, the generator G tries to minimise $W(\mathbb{P}_r, \mathbb{P}_g)$ between distributions $\mathbb{P}_r$ and $\mathbb{P}_g$, while the critic D tries to maximise this distance along with minimising the constraint $GP(\mathbb{P}_{\mathbf{y}_\mathbf{x}})$. The objective function of the WCGAN with gradient penalty (WCGAN-GP) is presented in equation (5.4), where λ_1 is the penalty coefficient. We use $\lambda_1 = 10$.

$$\begin{aligned}\mathcal{L}_{\text{WCGAN-GP}}(G, D) = &\arg\min_G \max_D W((\mathbb{P}_r)_{\mathbf{y}_\mathbf{x}\sim\mathbb{P}_r}, (\mathbb{P}_g)_{\widehat{\mathbf{y}}_\mathbf{x}\sim\mathbb{P}_g}) \\ &+ \lambda_1 \min_D GP(\mathbb{P}_{\mathbf{y}_\mathbf{x}}).\end{aligned} \tag{5.4}$$

We additionally impose an ℓ_1 loss to enforce the reconstruction to align with the ground truth. Our final objective function $\mathcal{L}$ is presented in equation (5.6).

$$\mathcal{L}_{\ell_1}(G) = \mathbb{E}_{\widehat{\mathbf{y}},\mathbf{y}}(\|\mathbf{y} - \widehat{\mathbf{y}}\|_1), \tag{5.5}$$

$$\mathcal{L} = \mathcal{L}_{\text{WCGAN-GP}}(G, D) + \lambda\mathcal{L}_{\ell_1}(G). \tag{5.6}$$

The hyperparameter λ is set to 100 after fine-tuning. The number of critic iterations per generator iteration is set to 2.

Latent Convolutional Transformer Layers (CTLs)

Inspired by 2D compact transformers [241], we implement our 3D latent convolutional Transformer layer (CTL) block that uses convolutional embeddings following transformer layers in the latent space. Since our data contain a large empty background representing non-vessel regions, the usual patch embeddings are not suitable, while in the latent space, the transformer can help us extract the relations between feature maps and enforce their importance via attention modules. We do not require positional embeddings for feature maps as they are order invariant. The latent CTLs consider the max pooling results $f_{\text{input}} \in \mathbb{R}^{N\times C\times H\times W\times D}$ from the last layer as input, where N is the batch size, C denotes the number of channels, and H, W, D stand for height, width, and depth, respectively. We next perform convolutional embeddings on the latent feature maps $f_{\text{embeddings}} \in \mathbb{R}^{N\times(H\times W\times D)\times C} = reshape(Conv_{1\times1\times1}(f_{\text{input}}))$. The feature map embeddings then go through the transformer layers as follows:

$$(Q, K, V) \in \mathbb{R}^{3\times N\times(H\times W\times D)\times C} = reshape(\text{LinearLayer}(\text{LayerNorm}(f_{\text{embeddings}}))), \quad (5.7)$$

$$\mathbf{z}' \in \mathbb{R}^{N\times(H\times W\times D)\times C} = \text{LayerNorm}(\text{MHSA}(Q, K, V) + f_{\text{embeddings}}), \quad (5.8)$$

$$\mathbf{z} \in \mathbb{R}^{N\times(H\times W\times D)\times C} = \text{LinearLayer}(\text{GELU}(\text{LinearLayer}(\mathbf{z}'))) + \mathbf{z}', \quad (5.9)$$

where MHSA denotes a multi-head self-attention module, GELU the activation layer of the Gaussian error linear unit, $\mathbf{z}'$ the intermediate result after MHSA, $\mathbf{z}$ the output of one transformer layer, and Q, K, V stand for query, keys, and values vectors, respectively. We use 8 attention heads and 8 transformer layers, where the output $\mathbf{z}$ for one layer will replace the $f_{\text{embeddings}}$ for the next layer.

After using the transformers to encode the relations between the latent feature maps, we use a linear layer to do final mappings on these embedded feature maps and reshape the

results as the same size as input.

$$f_{\text{output}} \in \mathbb{R}^{N\times C\times H\times W\times D} = reshape(\text{LinearLayer}(reshape(\text{LayerNorm}(\mathbf{z})))). \tag{5.10}$$

Dynamic Snake Convolutional (DSConv) Critic

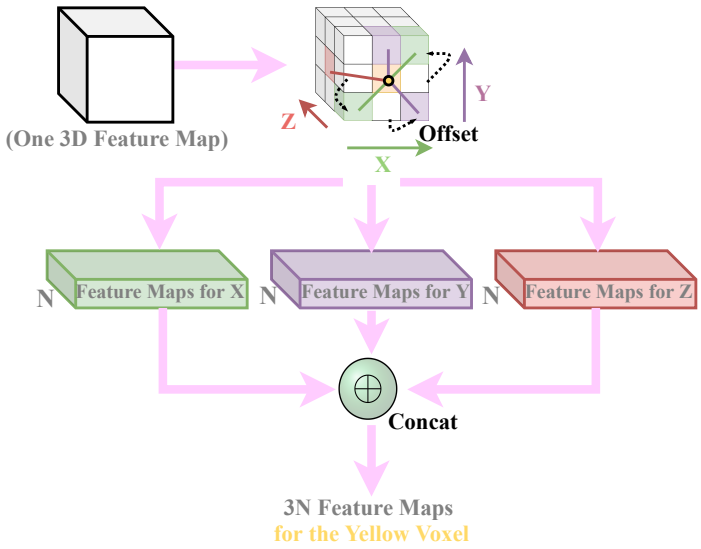


Figure 5.3: Dynamic snake convolution (DSConv). For each voxel (yellow voxel in the figure as an example) in the feature map, the DSConv flattens the whole kernel along different axes with random offsets to extract different dynamic feature maps for X-, Y-, and Z-axes separately and then concatenates these feature maps together as the convolution output.

The coronary tree is formed of quasi-tubular topological structures, and the traditional convolutional kernel is not optimal for recognising thin local structures and variable global morphologies. DSConv kernels [50] are designed specifically for such structures. Differing from the traditional 3D kernel with size $3 \times 3 \times 3$, DSConv flattens the whole kernel along different axes with random offsets to generate a dynamic snake-shaped kernel as illustrated in figure 5.3. For 3D data, DSConv generates three dynamic feature maps for X-, Y-, and Z-axes respectively and then concatenates these three feature maps together as the output. Due to its nature of dynamic offsets, DSConv can extract tubular features more efficiently. We use the DSConv in the first layer of the critic to extract global tubular features and concurrently perform traditional convolution to extract essential local features as well. We then concatenate these global and local features together, and apply downsamplings to

calculate the critic loss. This design of the critic can effectively distinguish vessel tubular structures.

5.3 Experimental Settings

5.3.1 Datasets

We use a public CCTA dataset [222] containing 3D binary segmented coronary trees for our study, and split the coronary trees into RCA and LAD. We use 879 segmented RCA data in total, dividing them into 75% training, 15% validation, and 10% test datasets. We perform the cone-beam forward projection using the TIGRE toolbox [220]. The volume size is $128 \times 128 \times 128$, and the detector size is 512×512. Details of other projection geometry parameters are provided in table 5.1.

We also collect a clinical ICA dataset of 8 patients for evaluation, who were admitted at the Oxford John Radcliffe Hospital with suspected coronary stenosis and provided informed consent. For two of the patients, three ICA projections were captured, while for the rest, there were two ICA projections. Our model takes 3D backprojection from binary projections as input, so these clinical ICA data are pre-segmented and then backprojected before evaluation.

5.3.2 Baseline Models and Implementation Details

As there is no equivalent previous work on this problem using deep learning, we implement four models as baselines for comparative analyses. We replace the 3D U-Net in WCGAN-GP with Unet++ [242] (termed as Un2+), with Unet+++ [243] (termed as Un3+), and with DSConv Network [50] (termed as DSCN). We also implement the 3D convolutional vision transformer GAN [244] (termed as CVTG). We use Adam optimiser for both generator and critic [236], with an initial learning rate of 10^{-4}. The training was performed with a batch size of 3 on NVIDIA Quadro RTX 8000.

5.3.3 Metrics

We adopt the overlap using a sweeping distance threshold ($Ot(d)$) as an evaluation metric, where d is the distance threshold in mm unit [202]. $Ot(d) \in [0, 1]$ with 0 representing no overlap and 1 the perfect match; the metric is equivalent to the *Dice* score when $d = 0$. The different d values allow us to measure reconstructions under different degrees of deformation. In addition, we use the Chamfer ℓ_2 distance (CD_{ℓ_2}) for measuring the corresponding voxel-wise or pixel-wise prediction errors (mm) in either 3D or 2D data according to their voxel or pixel spacing.

We evaluate the models on both the CCTA test dataset and the unseen real clinical ICA dataset. For the CCTA test dataset, we directly validate the results in 3D space after rigidly registering the ground truth to the predicted reconstruction using $Ot(d)$ with $d = \{1, 2\}$ mm and CD_{ℓ_2}. Since the training ground truth is the original CCTA data used to generate the first projection, there is no motion between the original ICA data and the reprojections of the predicted reconstructions on the first projection plane, while it exists on other projection planes. For this reason, we measure the *Dice* score between the ICA data and reprojections on the first projection plane (same as $Ot(0)$). For the second and any additional projection planes, we first rigidly register the ICA data to the reprojections. We then compute the $Ot(d)$ with $d = \{1, 2\}$ mm and CD_{ℓ_2} between them. All the 3D reconstructions on the CCTA test dataset and real clinical ICA dataset are binarised with a threshold of 0.5 before evaluation, reprojection, and visualisation. All the 2D reprojections are binarised with a threshold of 0 before evaluation and visualisation.

5.4 Results and Discussion

5.4.1 Analysis on 3D CCTA Test Dataset

As demonstrated in table 5.2, our proposed DeepCA model achieves the best performance in all metrics on the CCTA test dataset compared to the 4 baseline models. This indicates our model can better capture the vessel topological structures in the source domain of the

CCTA data.

Table 5.2: Quantitative performance of our proposed DeepCA model and 4 baseline models in terms of $Ot(d)$ (%) and CD_{ℓ_2} (mm). The *Dice* score (%) is equivalent to $Ot(0)$ (%). Best results are annotated in **bold**.

Model	3D CCTA Test Dataset			2D Real Clinical ICA Dataset (Unseen Domain)						
				1^{st}	2^{nd} Projection			Additional Projection		
	$Ot(1)\uparrow$	$Ot(2)\uparrow$	$CD_{\ell_2}\downarrow$	$Dice\uparrow$	$Ot(1)\uparrow$	$Ot(2)\uparrow$	$CD_{\ell_2}\downarrow$	$Ot(1)\uparrow$	$Ot(2)\uparrow$	$CD_{\ell_2}\downarrow$
Un2+ [242]	51.99 (±12.17)	64.75 (±13.28)	4.64 (±2.01)	65.42 (±6.68)	22.97 (±10.82)	31.05 (±12.76)	12.79 (±4.61)	25.25 (±18.40)	39.37 (±20.75)	8.49 (±3.26)
Un3+ [243]	55.10 (±10.72)	68.69 (±10.96)	4.49 (±1.67)	62.23 (±9.49)	22.27 (±8.30)	30.19 (±10.51)	12.20 (±3.77)	32.92 (±23.99)	42.37 (±26.30)	7.70 (±1.59)
DSCN [50]	61.74 (±14.28)	72.21 (±13.31)	3.49 (±1.68)	**83.59** (±4.01)	30.66 (±11.30)	42.74 (±11.68)	7.81 (±2.19)	53.26 (±6.74)	67.90 (±4.77)	3.92 (±0.48)
CVTG [244]	61.53 (±11.49)	73.71 (±10.75)	3.51 (±1.36)	76.84 (±5.73)	32.98 (±12.29)	44.85 (±15.85)	8.23 (±3.73)	49.50 (±0.82)	64.63 (±3.05)	4.22 (±0.51)
DeepCA	**64.21** (±10.78)	**76.25** (±9.72)	**3.22** (±1.20)	83.31 (±4.32)	**45.70** (±6.79)	**58.39** (±8.42)	**4.51** (±1.29)	**58.58** (±4.01)	**72.88** (±0.90)	**2.81** (±0.06)

All values represent mean (± standard deviation).

We also visualise the corresponding voxel-wise prediction errors in terms of CD_{ℓ_2} between the ground truth and 3D predictions, as illustrated in figure 5.4. We can see that our proposed model can reconstruct all the branches, though the reconstructed sinoatrial nodal arteries (marked by the black boxes) have larger offsets compared to the ground truth due to deformations. More qualitative results on 3D CCTA test dataset are illustrated in appendix B.

5.4.2 Analysis on 2D Clinical ICA Dataset

From the quantitative results presented in Table 5.2, we can observe that our proposed DeepCA model attains the best performance on real ICA data in all metrics on both the second and additional projection planes compared to the 4 baseline models. For the first projection plane, our model achieves the second best performance, with only 0.33% behind the DSCN in terms of *Dice* score. However, in the second and additional projection planes, which are mostly affected by motion, our method performs the best. In particular, the results for the additional projection plane are the most significant, since the model is trained only based on two projections. Our proposed model presents large improvements

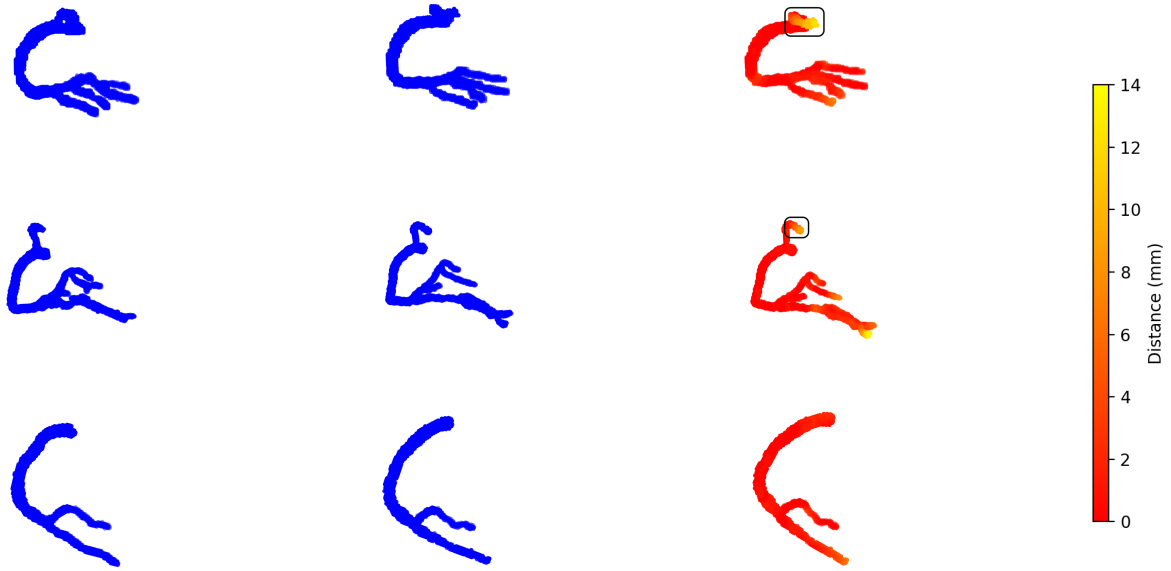


Figure 5.4: Three 3D reconstruction results on the CCTA test dataset by our DeepCA model. From top to bottom: three CCTA samples. From left to right: predicted reconstruction, ground truth, and corresponding voxel-wise prediction errors in terms of CD_{ℓ_2}.

of 38.57% and 9.99% in terms of $Ot(1)$, 42.25% and 28.32% in terms of CD_{ℓ_2}, for the second and additional projection planes respectively, compared to the best baseline models. This indicates our model has a better generalisation ability to the unseen domain.

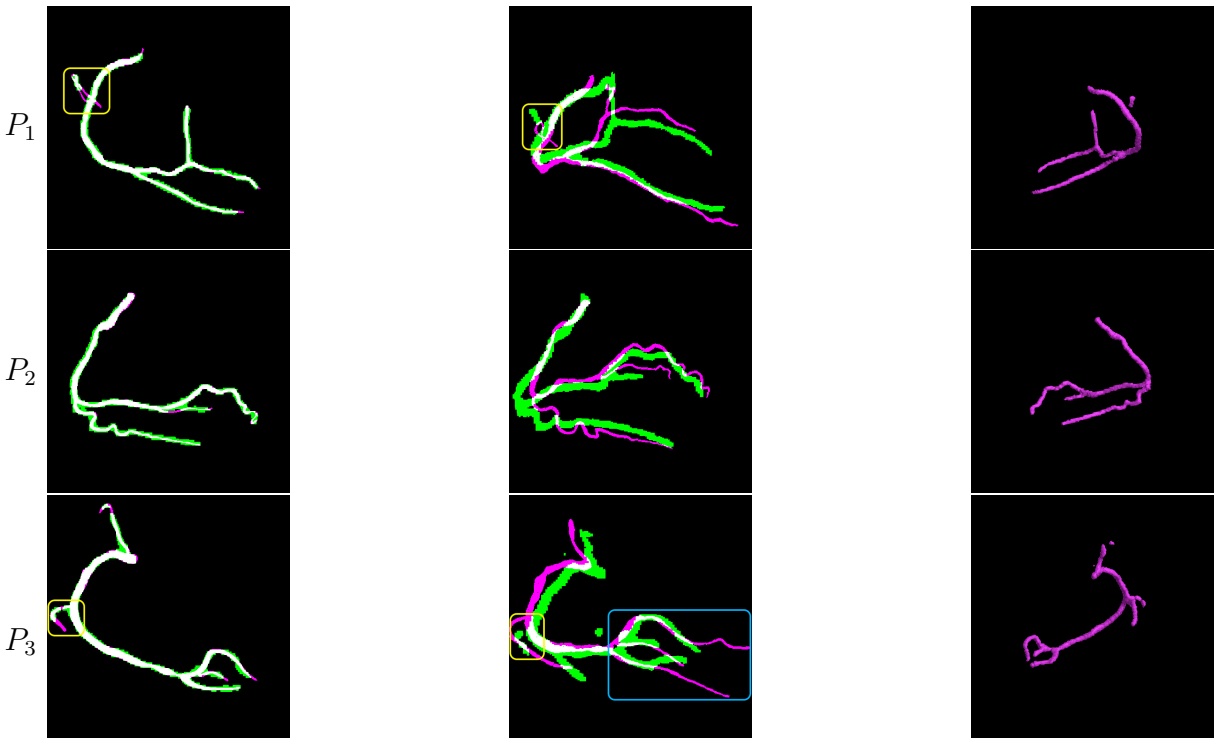
An interesting qualitative example is presented in figure 5.5, where we observe a missing vascular section at the middle of the main RCA branch in the original ICA data, as marked by the pink box. This section is successfully recovered during 3D reconstruction, demonstrating our model's generative ability to recover missing vascular structures that may be missing due to acquisition or panning/zooming errors.

Figure 5.6 shows three qualitative results by our proposed DeepCA model on real ICA data. We find that our model can reconstruct almost all the branches, especially the posterior descending arteries. Overall, the results illustrate good vessel connectivity, though there exist some broken acute marginal branches as marked by the yellow boxes in figure 5.6. The reason behind this may be that those areas are affected heavily by motion, causing incomplete reconstruction. We also note that in the second projection of P_3 in figure 5.6,

Original ICA 3D Reconstruction

Figure 5.5: An example of 3D reconstruction for the RCA branch of a patient. Left: original ICA data. Right: 3D reconstruction by our proposed DeepCA model.

the reconstructed posterior descending arteries are shorter as marked by the blue box. This may be the result of non-simultaneous ICA acquisitions where the contrast agent arrives at different distances between projections causing vessel differences in the different angiographic scans.



1^{st} Projection 2^{nd} Projection 3D Reconstruction

Figure 5.6: Three qualitative examples. From top to bottom: three patients $P_{1,2,3}$. From left to right: comparisons between the real ICA data and our reprojections on the first and second projection planes after rigid registration, and our DeepCA model's 3D reconstruction result. The colour purple represents ICA data, green represents reprojection, and white shows the overlap.

Figure 5.7 shows two example evaluations of our DeepCA model's 3D reconstruction on an

additional projection plane. It demonstrates that even without involving the information of this projection plane in the input, our model can still reconstruct the accurate vascular structures. More qualitative results on the clinical ICA data are illustrated in appendix B.

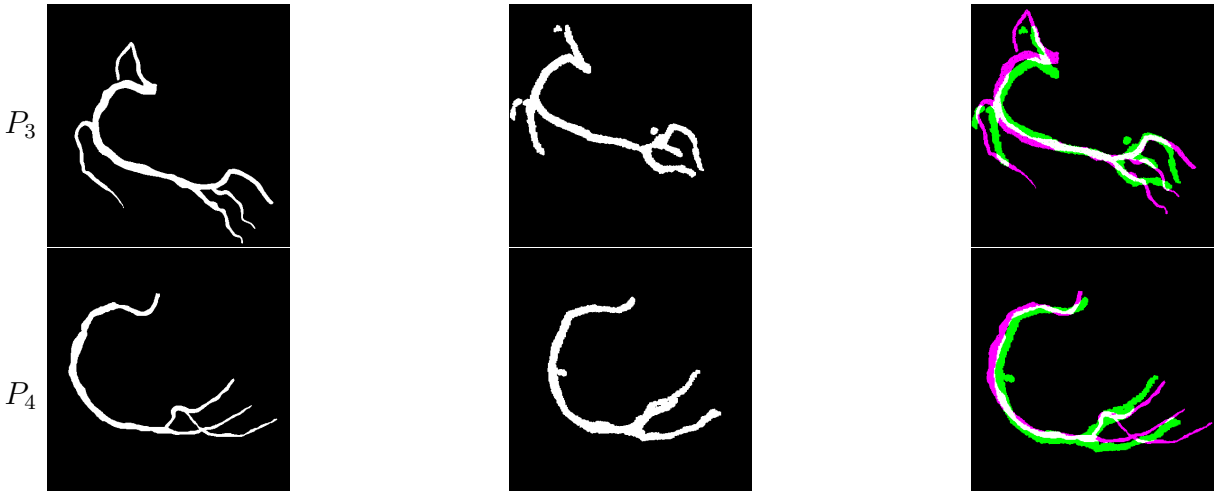


Figure 5.7: Two example cases of our DeepCA model's reconstruction on the additional projection plane. From top to bottom: two patients $P_{3,4}$. From left to right: original ICA data, reprojection, and comparison between them after rigid registration. The colour purple represents ICA data, green represents reprojection, and white shows the overlap.

5.4.3 Ablation Study

We evaluate the significance of different components of our proposed DeepCA model through an ablation study. We evaluate (1) WCGAN-GP (termed as WGP), (2) WCGAN-GP with latent CTLs (termed as +CTLs), and (3) WCGAN-GP with DSConv critic (termed as +DSCC). As demonstrated in Table 5.3, the combined models consistently provide improved performance. The qualitative results in the supplementary material on real ICA data also illustrate the advantages of each component of our model.

Table 5.3: Quantitative results of 3 ablation models in terms of $Ot(d)$ (%) and CD_{ℓ_2} (mm). Best results are annotated in **bold**.

Model	3D CCTA Test Dataset		
	$Ot(1) \uparrow$	$Ot(2) \uparrow$	$CD_{\ell_2} \downarrow$
WGP	$62.06_{\pm 10.61}$	$74.38_{\pm 10.01}$	$3.43_{\pm 1.29}$
+CTLs	$62.87_{\pm 11.68}$	$74.39_{\pm 10.70}$	$3.24_{\pm 1.23}$
+DSCC	$63.46_{\pm 10.85}$	$75.14_{\pm 10.00}$	$3.24_{\pm 1.23}$
DeepCA	$\mathbf{64.21}_{\pm 10.78}$	$\mathbf{76.25}_{\pm 9.72}$	$\mathbf{3.22}_{\pm 1.20}$

All values represent mean (± standard deviation).

5.5 Conclusion

In this chapter, we propose DeepCA, leveraging the WCGAN with gradient penalty, latent convolutional transformer layers, and a dynamic snake convolutional critic for accurate 3D coronary artery tree reconstruction. Through simulating projections from CCTA data, we achieve generalisation on real non-simultaneously acquired ICA data. We use a special metric designed for the problem together with Chamfer ℓ_2 distance and validate our proposed model on both a CCTA dataset and a real ICA dataset. Both the quantitative and qualitative results demonstrate the promising performance of our proposed DeepCA model in vessel topology preservation, recovery of missing features, and generalisation ability. To the best of our knowledge, this is the first study that leverages deep learning in 3D coronary tree reconstruction from two real non-simultaneous ICA projections. The evaluations in this paper provide a baseline for future work in this area.

Chapter 6

Iterative Motion Compensation

Abstract - IterCA: Deep Iterative Motion Compensation for 3D Coronary Artery Tree Reconstruction from Two 2D Non-simultaneous Projections

DeepCA in chapter 5 successfully tackles the 3D coronary artery tree reconstruction problem based on non-simultaneous projections, but it presumes a certain level of motion in real ICA data and has not been evaluated extensively under various scenarios. In this chapter, a novel iterative pipeline is proposed, namely IterCA, based on DeepCA to explicitly compensate for motion via iterative registration on the non-simultaneous projection plane, in order to refine the 3D coronary artery tree reconstruction. Four different misalignment levels are simulated based on a public CCTA dataset to reasonably approximate typical misalignment amounts found in clinical acquisitions for training. The trained model evaluated the best across different misalignment levels is selected for better generalisability under various unseen real-world scenarios. The proposed IterCA pipeline is validated on the CCTA test dataset and three unseen datasets. The results demonstrate that IterCA achieves accurate and stable performance in vessel topology preservation, branch-connectivity maintaining, and generalisation ability to real ICA data and data from unseen domains.

6.1 Introduction

Vessel misalignment between non-simultaneous scans because of cardiac and respiratory motions worsens the 3D reconstruction quality. DeepCA [245] proposed in chapter 5, at the time of publication, is the only deep leaning-based approach to the task of 3D coronary artery tree reconstruction from two non-simultaneous projections. But it assumes a limited

level of motion in real ICA data during acquisitions and the model has not been evaluated extensively under different scenarios. AutoCAR [177] is the latest deep learning-based approach to the task of sparse 3D dynamic cardiovascular reconstruction, but whether it can handle cardiac motion between projections is unknown and not explicitly stated. In chapter 5, we show that the simulated 3D deformation can approximate the clinical motion and movement to enable generalisation on real data, so we directly use the term 'motion/movement' for the simulation. In this chapter, we explore the effect of different simulated deformations on generalisability, with an emphasis on how they cause different severity levels of the vessel misalignment between two 2D projection planes and how these different vessel misalignment levels impact generalisability. Therefore, we change the terminology to 'misalignment' as a more straightforward description of vessel misalignment between non-simultaneous scans because of motion and movement.

In this chapter, we propose a novel iterative pipeline, IterCA, to explicitly compensate for rigid and non-rigid motions via iterative registration on the non-simultaneous projection plane, in order to refine the 3D coronary artery tree reconstruction. Our pipeline is based on the revised DeepCA model trained on a public CCTA dataset, which is augmented with four levels of misalignment to approximate typical misalignment amounts found in clinical acquisitions. We simulate 2D projections on two different projection planes from CCTA data, with different levels of rigid transformation introduced to the CCTA data before forward projection on the second projection plane, and use the two simulated projections to learn from the CCTA data. We then choose the trained model with the best performance across all misalignment levels to enable better generalisation to various real-world scenarios. In this way, we overcome the problems of both the limited number of real paired ICA data with projection geometry information and the unavailable 3D ground truth for real ICA data. Next, we perform forward projection from the 3D reconstruction results on the second projection plane and register the original ICA image to the reprojection image to explicitly correct the motion. After that, the registered image is regarded as the new second ICA image that is combined with the original first ICA image and is sent to the

model again to produce the next improved reconstruction. We run this process iteratively to refine the 3D coronary tree reconstruction until the predefined criteria are satisfied. We focus on the RCA in this study, because RCA undergoes more compressive strain and is affected more by motion artifacts than other coronary vessels. We validate our proposed pipeline on the CCTA test dataset and three unseen datasets including an unseen CCTA dataset, a synthetic dataset, and a real ICA dataset. We provide an application-specific evaluation method to tackle the deformation in 3D reconstructions, the unavailability of 3D ground truth for real ICA scans, and the motion between projection planes, together with Chamfer ℓ_2 distance. The extensive evaluation results demonstrate our proposed pipeline achieves promising stable performance in vessel topology preservation, branch-connectivity maintaining, and generalisation ability to real non-simultaneous ICA data compared to the DeepCA model. Our main contributions are:

1. **3D coronary artery tree reconstruction from two non-simultaneous projections:** By training a deep learning model on two simulated projections from CCTA data, we achieve 3D coronary artery tree reconstruction from two real non-simultaneous ICA projections.

2. **Generalisation:** We propose four groups of misalignment levels to reasonably approximate general misalignment amounts found in clinical acquisitions to train our models, which achieves better generalisation to various unseen real-world scenarios.

3. **Iterative motion compensation:** Based on the trained deep learning model, we explicitly compensate for rigid and non-rigid motions via iterative registration on the non-simultaneous projection plane to refine the 3D coronary artery tree reconstruction.

4. **Extensive evaluation:** We use a specific metric designated for this problem in the absence of motion-free 3D ground truth and perform thorough evaluation for our proposed pipeline on the CCTA test dataset and three unseen datasets, which provides a strong baseline for future quantitative comparisons.

6.2 Materials and Methods

Our proposed pipeline IterCA consists of three stages: misalignment simulation, model training, and inference process, as illustrated in figure 6.1, where stages (b) and (c) constitute our proposed deep iterative motion compensation. At the first stage, we simulate the real ICA projections by generating two projections from 3D CCTA data, with different levels of simulated misalignments on the second projection plane. We then apply backprojection on the two simulated projections to produce input to the model at the next step. Next, we map the 3D backprojection input to the CCTA data for learning 3D coronary artery tree reconstruction via training deep neural networks, implicitly compensating for any motion. We select the model which obtains the best performance and freeze it. Finally, based on this frozen model, we iteratively compensate for motion via registration on the second non-simultaneous projection plane to refine the 3D coronary tree reconstruction.

6.2.1 Proposed Misalignment Simulation

Due to the lack of 3D ground truth for real ICA data, we use a public CCTA data set [222] containing real 3D segmented coronary trees, and simulate projections based on them for our study. Breathing and cardiac motions introduce deformations to the coronary tree between non-simultaneous projections and it is found [246] that rigid transformations could be good approximations for real conditions when using deep learning. Therefore, to resemble real non-simultaneous ICA projections, we simulate 2D projections on two different planes from CCTA data, with a rigid transformation applied to the CCTA data before forward projection on the second projection plane. Since contrast injections used in real ICA projections change image intensity values between projections, the ICA images are pre-segmented, where points on vessels are assigned as 1 and background as 0. Hence, in order to generalise to real ICA projections, we binarise the simulated projections with a threshold of 0 as well, i.e., any points with values greater than 0 are set to 1 and 0 otherwise. We use the projection geometry of real coronary angiography to simulate

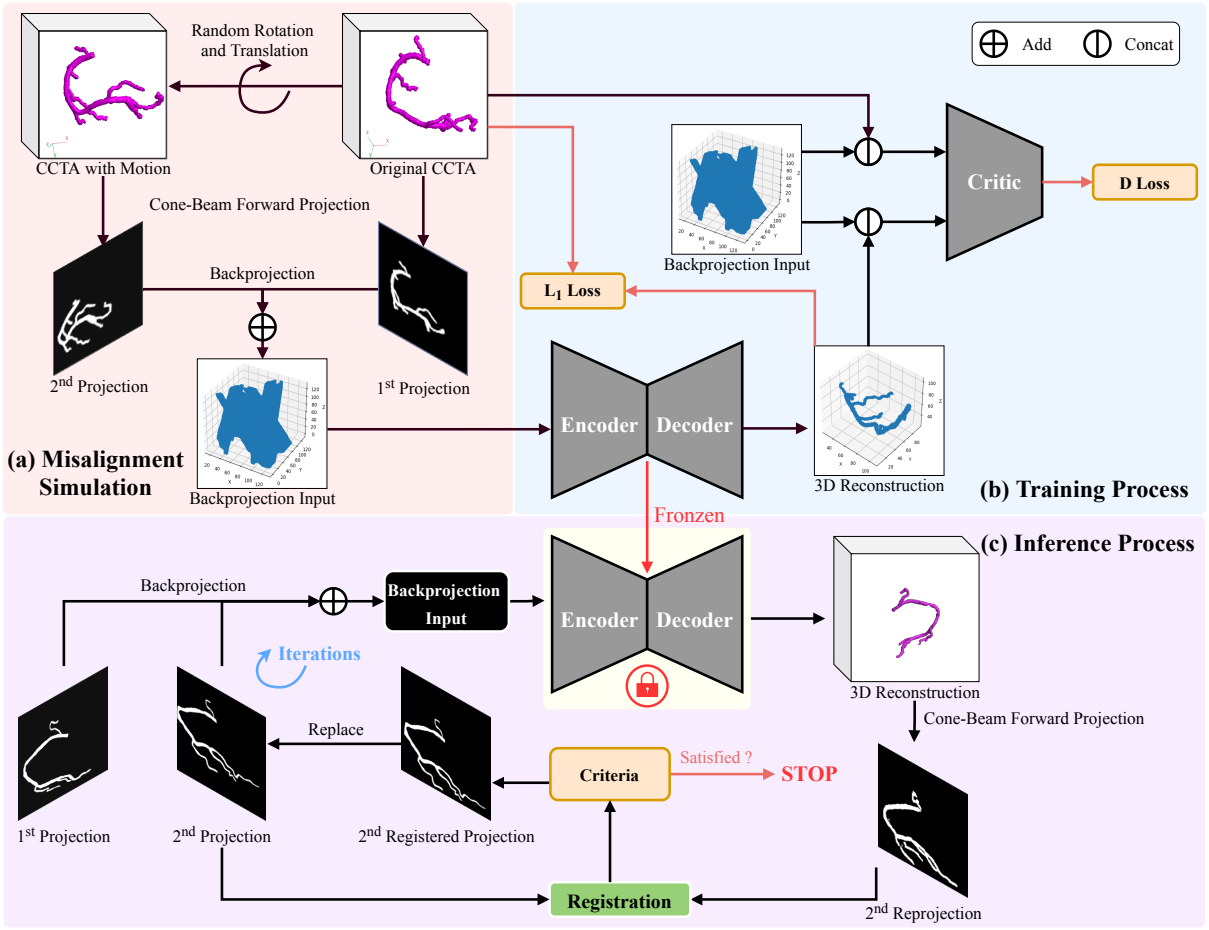


Figure 6.1: Graphical summary of our proposed IterCA pipeline. **(a)** We generate two simulated projections from CCTA data, including different simulated misalignments between projections, and then produce the 3D model input via performing backprojection on the two simulated projections. **(b)** Our model with a conditional generator and a critic receives the 3D backprojection as input to learn from the CCTA data for 3D coronary tree reconstruction, implicitly compensating for motion. **(c)** Based on the frozen trained model, we iteratively compensate for motion via registration on the second non-simultaneous projection plane to refine the 3D coronary tree reconstruction during inference.

the two cone-beam forward projections. Details of projection geometry parameters are presented in table 5.1 in chapter 5.

To enable better generalisation to various real-world scenarios, we design a three-step process to synthesise training datasets simulated with the proposed four levels of misalignment. First, we regard the first projection plane as the reference plane, so we directly perform cone-beam forward projection of the original CCTA data on this plane. Second, before forward projection on the second projection plane, we introduce rigid transformations as misalignments to the CCTA data. We sample parameters from a normal distribution

with zero mean for translations along the X-, Y-, and Z-axes, as well as rotations of both primary and secondary angles. According to [246], different simulated misalignments to training data can have an impact on the final results. We do not presume a certain misalignment range like [245]. Instead, based on the analysis of the misalignment levels affected by cardiac and respiratory motions in real clinical scenarios [247–250], we propose four individual groups, represented by normal distributions, as different misalignment levels with increasing standard deviation values for each, namely, mild, medium, strong and severe, as demonstrated in table 6.1. For each CCTA data, we apply 10 different sets of random misalignments sampled from the given normal distribution and this finally results in four datasets with mild, medium, strong and severe simulated misalignments. Finally, we apply backprojection to the two binary simulated projections using the same known projection geometry separately to transform the projection information back into 3D space and binarise the two 3D backprojections with a threshold of 0. We add them together to generate a single 3D input to our models, which learns from the original CCTA data, allowing us to train our models under realistic conditions, so enabling generalisation to real non-simultaneous ICA projections.

Table 6.1: Misalignment amounts per severity level expressed as translations (mm) along (x, y, and z) axes and rotations (°) of both primary and secondary angles. For each misalignment level, the values are standard deviation values of normal distributions with zero mean.

Level:	Mild	Medium	Strong	Severe
Translations (mm)	(2.5, 2, 5)	(5, 4, 10)	(7.5, 6, 15)	(10, 8, 20)
Rotations (°)	2.5	5	7.5	10

We use 669 RCA samples in total from the CCTA dataset [222] and divide them into 75% training, 15% validation, and 10% test datasets. We then synthesise them with four levels of misalignment into four datasets, obtaining a total of $669 \times 10 = 6,690$ data under each severity level to train each model.

Test datasets

In addition to the four CCTA test datasets generated with different misalignment levels, we also generate a CCTA test dataset without any misalignments as the non-misalignment group, so we have in total five CCTA test datasets. We evaluate our method on three more unseen datasets: a new CCTA dataset from a different centre (termed as CCTA-UNSW) [224], a synthetic RCA dataset generated by a vessel generator (termed as VG) [171], and a real ICA dataset collected from the Oxford John Radcliffe Hospital. The CCTA-UNSW dataset contains 40 CCTA data, the VG dataset contains 80 synthetic RCA data, and the real ICA dataset contains 8 data from 8 patients with provided informed consent. For both CCTA-UNSW and VG datasets, we simulate the same four misalignment and one non-misalignment level for testing. In the real ICA dataset, three ICA projections were captured for two of the patients, whereas for the rest, there were two ICA projections. Our model takes 3D backprojection from two binary projections as input, so these clinical ICA data are pre-segmented and then backprojected before evaluation.

6.2.2 Proposed Deep Iterative Motion Compensation

Backbone Model - DeepCAv2

The backbone model of our pipeline is based on a deep learning model revised from DeepCA [245] leveraging the WCGAN with gradient penalty, CTLs, and a DSConv critic. Each component has been proven to be effective on the task. Here we do not add paddings when performing the latent convolutional embeddings, because the embeddings are performed in deep-layer feature maps in the latent space.

Most motion artifacts are corrected by our models via mapping the 3D backprojection input with non-aligned projections to 3D CCTA data. We systematically analyse the performance of our models trained with different groups of misalignments separately and select the model (named as DeepCAv2) which achieves the best performance across all different severity levels, in order to enable better generalisation to various unseen real-world

scenarios.

Iterative Inference Approach

The heart at different cardiac phases is transformed by different rotations, translations, and contraction compared to the reference heart model at diastole, due to heart beating. Since we simulate the misalignments with only rigid transformations, there is still residual motion affecting the 3D reconstruction by our model DeepCAv2. Therefore, we propose a novel pipeline, named IterCA, to iteratively refine the 3D coronary artery tree reconstruction and explicitly compensate for residual motion.

The ground truth is the original CCTA data which we used to generate the first projection, so there is no motion effect on the first projection plane whereas it exists on the other projection planes. We perform forward projection of the predicted 3D reconstruction produced by our DeepCAv2 model, following the same second projection geometry to obtain a predicted second reprojection. We then register the original second ICA projection to the reprojection to explicitly correct for motion. Finally, we use the registered second ICA projection with the original first ICA projection to repeat the backprojection and go through the same steps iteratively. DeepCAv2 model is frozen during iterative refining. We use rigid registration at the first iteration to correct the initial large displacement. Specifically, we regard the 2D reprojection and ICA data as point clouds, and then leverage iterative closest point (ICP) [251] to find their correspondences and iteratively minimise the distance between them to calculate a rigid transformation for registration. We use distance-based rejection threshold in ICP to exclude poor matches [252]. This enables a stable, robust, and fast initial registration. In the following iterations after an initial large displacement is registered, we change into non-rigid registration for a fine structural contour registration. The non-rigid registration is based on the 'Demons' algorithm [253, 254] that estimates the displacement field to iteratively map the moving image onto the reference image. We set stop criteria before the start of each iteration to measure whether optimal results are met for different data. One criterion is the maximum number of iterations and

another one is the reconstruction improvement based on the second projection plane, as illustrated in algorithm 1.

Data: first projection $\mathbf{p_1}$, second projection $\mathbf{p_2}$, frozen model $\mathcal{M}$ after training, maximum number of iterations $N = 15$, and criteria threshold $\epsilon = 0.65$
Result: refined 3D coronary artery tree reconstruction $\mathbf{Y}$
$n \leftarrow 1$;
$t \leftarrow 0$;
while $n \leq N$ ***and*** $t < \epsilon$ **do**
  $\mathbf{x} \leftarrow \text{Bin}_0(\text{BP}(\mathbf{p_1})) + \text{Bin}_0(\text{BP}(\mathbf{p_2}))$;
  $\mathbf{Y} \leftarrow \text{Bin}_{0.5}(\mathcal{M}(\mathbf{x}))$;
  $\widehat{\mathbf{p}}_\mathbf{2} \leftarrow \text{Bin}_0(\text{FP}_2(\mathbf{Y}))$;
  if $n = 1$ **then**
    $\mathbf{p}'_\mathbf{2} \leftarrow \text{Reg}_{\text{rigid}}(\mathbf{p_2}, \widehat{\mathbf{p}}_\mathbf{2})$;
  else
    $\mathbf{p}'_\mathbf{2} \leftarrow \text{Reg}_{\text{non-rigid}}(\mathbf{p_2}, \widehat{\mathbf{p}}_\mathbf{2})$;
  end
  $\mathbf{p_2} \leftarrow \text{Reg}_{\text{rigid}}(\mathbf{p_2}, \widehat{\mathbf{p}}_\mathbf{2})$;
  $ot \leftarrow Ot(1)(\mathbf{p_2}, \widehat{\mathbf{p}}_\mathbf{2})$;
  $cd \leftarrow CD_{\ell_2}(\mathbf{p_2}, \widehat{\mathbf{p}}_\mathbf{2})$;
  $t \leftarrow ot - \frac{cd}{100}$;
  $n \leftarrow n + 1$;
  $\mathbf{p_2} \leftarrow \mathbf{p}'_\mathbf{2}$;
end

Algorithm 1: Iterative motion compensation via 2D Registration. BP is backprojection, FP_i is cone-beam forward projection on the i_{th} plane, Bin_x is binarisation with a threshold of x, $\text{Reg}_{\text{(rigid or non-rigid)}}(m, f)$ is to register image m to image f, $Ot(1)(m, f)$ is to measure the deformable overlap between images m and f (see section 6.2.4), and $CD_{\ell_2}(m, f)$ is to measure the corresponding prediction errors between images m and f (see section 6.2.4).

6.2.3 Training Setup

We use the deep learning model DeepCA [245] as the baseline and follow the same setup to train our model under four misalignment levels separately. We perform the cone-beam forward projection using the TIGRE toolbox [220]. We implement our model using PyTorch [235] and choose Adam optimiser [236] for both generator and critic, with a learning rate of 10^{-4} for both. We use stochastic weight averaging [255] to improve generalisation. The training was performed with a batch size of 3 on an HPC cluster utilising NVIDIA Quadro RTX 8000 48GB.

6.2.4 Evaluation Metrics

We adopt the overlap using a sweeping distance threshold ($Ot(d)$) as an application-specific evaluation metric, where d is the distance threshold in mm unit [202]. $Ot(d) \in [0, 1]$ with 0 representing no overlap and 1 the perfect match; this metric is equivalent to the Dice score when $d = 0$. The different d values allow us to compare vessels' similarity and structure under different degrees of deformation. In addition, we use the Chamfer ℓ_2 distance (CD_{ℓ_2}) for measuring the corresponding voxel or pixel-wise prediction errors (mm) according to their voxel or pixel spacing, which allows us to measure how much deformation is corrected.

For evaluation in terms of CCTA, CCTA-UNSW, and VG datasets, we directly validate the results in 3D space after rigidly registering the ground truth to the predicted reconstruction. Regarding clinical 2D ICA data, we perform evaluation between the ICA data and the reprojections on the first, second, and extra projection plane if available. On the first projection plane, we calculate $Ot(0)$ (same as Dice score) as it is a motion-free plane. For the second and possible extra projection planes, we first rigidly register the original ICA data to the reprojections and then perform evaluation. This fairly applies to our iterative approach as well though we use non-rigid registration during iterative refining. All the 3D reconstruction results are binarised with a threshold of 0.5 before evaluation, reprojection, and visualisation. All the 2D reprojections are binarised with a threshold of 0 before evaluation and visualisation.

6.3 Results

6.3.1 Performance of Models Trained with Different Misalignment Levels

We present box plots in figure 6.2 for the reconstruction performance of our models trained on CCTA datasets simulated with four different groups of misalignments and evaluated on CCTA test datasets with five different misalignment levels, in terms of $Ot(1)$ and CD_{ℓ_2}.

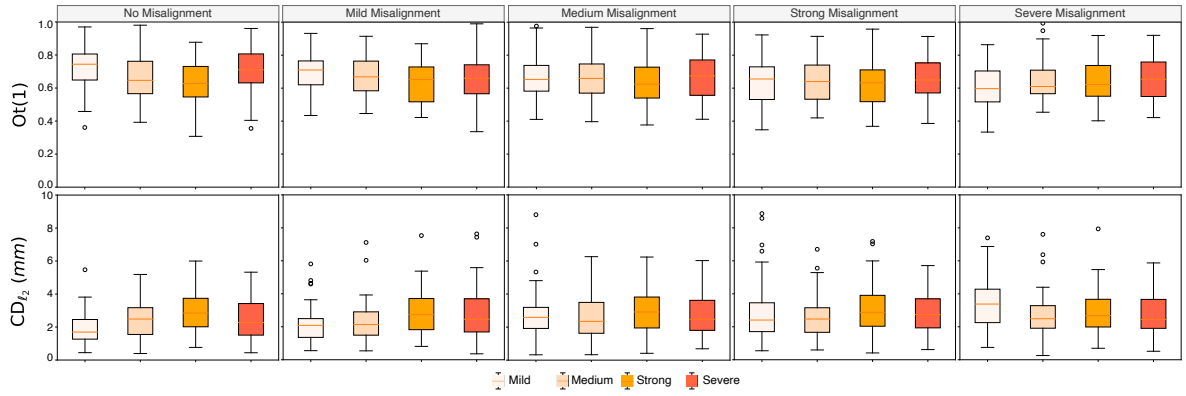


Figure 6.2: Box plots for the reconstruction performance of our models trained on CCTA datasets simulated with four different groups of misalignments (namely mild, medium, strong and severe presented in different coloured boxes) and evaluated on CCTA test dataset with five different misalignment levels (left to right), in terms of $Ot(1)$ and CD_{ℓ_2} (mm) (top to bottom).

According to the quantitative results in figure 6.2, we observe the model trained with simulated medium misalignments has the best overall and stable performance across all misalignment levels except for no and mild-misalignment groups. No misalignment is rare in real clinical settings, so we focus on the other four misalignment levels. For evaluation on the dataset with mild misalignments, the model trained with mild misalignments achieves the best performance, but its performance decreases significantly on datasets with increasing misalignments compared to the model trained with medium misalignments which is more stable across all. Therefore, we select the model trained with medium misalignments as DeepCAv2 model.

6.3.2 Performance of Iterative Motion Compensation Approach

Table 6.2 demonstrates the quantitative performance of our proposed pipeline IterCA, DeepCAv2, and DeepCA evaluated on three datasets including CCTA test dataset and two unseen datasets CCTA-UNSW and VG, with four misalignment levels and no misalignment, in terms of $Ot(1)$ and CD_{ℓ_2}. We do not test IterCA on datasets with no misalignment, since there is no motion between projections to correct iteratively, and IterCA's performance without iterations is same as DeepCAv2. DeepCAv2 achieves a large improvement compared to DeepCA for all 15 datasets and 2 metrics, except for 2 VG datasets with no and mild

misalignments. This proves the effectiveness of our proposed different misalignment exploration to choose the model with best generalisation to various scenarios, compared to DeepCA which assumes a certain level of motion during acquisitions. Our proposed pipeline IterCA further improves the reconstruction from DeepCAv2 on all datasets with all four misalignment levels in all metrics, showing that our iterative approach can better recover vessel topological structure and further compensate for any residual motion.

Table 6.2: Evaluation results of our proposed IterCA, DeepCAv2, and DeepCA evaluated on three datasets including CCTA test dataset, two unseen datasets CCTA-UNSW and VG, simulated with four misalignment levels (namely, mild, medium, strong, and severe) and no misalignment, in terms of $Ot(1)$ and CD_{ℓ_2} (mm). Best results are annotated in **bold**.

Misalignment Levels		None		Mild		Medium		Strong		Severe	
Dataset	**Method**	$Ot(1)\uparrow$	$CD_{\ell_2}\downarrow$	$Ot(1)\uparrow$	$CD_{\ell_2}\downarrow$	$Ot(1)\uparrow$	$CD_{\ell_2}\downarrow$	$Ot(1)\uparrow$	$CD_{\ell_2}\downarrow$	$Ot(1)\uparrow$	$CD_{\ell_2}\downarrow$
CCTA	DeepCA [245]	$0.63_{\pm 0.13}$	$2.84_{\pm 1.13}$	$0.58_{\pm 0.10}$	$3.28_{\pm 1.41}$	$0.57_{\pm 0.10}$	$3.61_{\pm 1.42}$	$0.56_{\pm 0.11}$	$3.71_{\pm 1.34}$	$0.54_{\pm 0.11}$	$4.21_{\pm 1.84}$
	DeepCAv2	$\mathbf{0.66}_{\pm 0.15}$	$\mathbf{2.43}_{\pm 1.18}$	$0.67_{\pm 0.12}$	$2.28_{\pm 1.17}$	$0.66_{\pm 0.13}$	$2.54_{\pm 1.30}$	$0.64_{\pm 0.12}$	$2.62_{\pm 1.27}$	$0.65_{\pm 0.12}$	$2.65_{\pm 1.32}$
	IterCA	N/A	N/A	$\mathbf{0.70}_{\pm 0.12}$	$\mathbf{2.08}_{\pm 1.08}$	$\mathbf{0.70}_{\pm 0.13}$	$\mathbf{2.16}_{\pm 1.04}$	$\mathbf{0.69}_{\pm 0.12}$	$\mathbf{2.27}_{\pm 1.14}$	$\mathbf{0.69}_{\pm 0.12}$	$\mathbf{2.21}_{\pm 0.99}$
CCTA - UNSW	DeepCA [245]	$0.52_{\pm 0.11}$	$4.37_{\pm 2.10}$	$0.51_{\pm 0.08}$	$4.75_{\pm 2.13}$	$0.49_{\pm 0.09}$	$5.20_{\pm 2.75}$	$0.44_{\pm 0.12}$	$6.25_{\pm 2.83}$	$0.44_{\pm 0.10}$	$6.53_{\pm 3.67}$
	DeepCAv2	$\mathbf{0.55}_{\pm 0.12}$	$\mathbf{4.32}_{\pm 2.06}$	$0.55_{\pm 0.11}$	$4.40_{\pm 2.00}$	$0.53_{\pm 0.10}$	$4.89_{\pm 2.28}$	$0.52_{\pm 0.12}$	$5.01_{\pm 2.38}$	$0.52_{\pm 0.09}$	$5.06_{\pm 2.97}$
	IterCA	N/A	N/A	$\mathbf{0.59}_{\pm 0.09}$	$\mathbf{3.71}_{\pm 1.66}$	$\mathbf{0.57}_{\pm 0.10}$	$\mathbf{4.13}_{\pm 2.12}$	$\mathbf{0.57}_{\pm 0.11}$	$\mathbf{4.35}_{\pm 2.34}$	$\mathbf{0.56}_{\pm 0.10}$	$\mathbf{4.14}_{\pm 2.04}$
VG	DeepCA [245]	$\mathbf{0.45}_{\pm 0.07}$	$\mathbf{5.53}_{\pm 1.58}$	$0.43_{\pm 0.08}$	$5.97_{\pm 1.65}$	$0.40_{\pm 0.10}$	$7.17_{\pm 2.84}$	$0.37_{\pm 0.11}$	$8.10_{\pm 3.66}$	$0.37_{\pm 0.11}$	$8.86_{\pm 4.14}$
	DeepCAv2	$0.44_{\pm 0.11}$	$6.39_{\pm 2.13}$	$0.44_{\pm 0.12}$	$6.36_{\pm 2.69}$	$0.42_{\pm 0.10}$	$6.81_{\pm 2.32}$	$0.41_{\pm 0.11}$	$7.22_{\pm 2.83}$	$0.38_{\pm 0.12}$	$8.21_{\pm 3.49}$
	IterCA	N/A	N/A	$\mathbf{0.48}_{\pm 0.10}$	$\mathbf{5.56}_{\pm 2.25}$	$\mathbf{0.46}_{\pm 0.09}$	$\mathbf{5.77}_{\pm 1.97}$	$\mathbf{0.47}_{\pm 0.09}$	$\mathbf{5.86}_{\pm 1.97}$	$\mathbf{0.47}_{\pm 0.10}$	$\mathbf{6.13}_{\pm 2.53}$

All values represent mean ± standard deviation.

We additionally show the quantitative performance of our proposed IterCA, DeepCAv2, and DeepCA on the unseen real ICA data in table 6.3. It shows gradual improvements from DeepCA, to DeepCAv2 and finally to IterCA on all three projection planes, which demonstrates the effectiveness of our proposed different misalignment simulation and iterative motion compensation in real scenarios. Our proposed IterCA method attains the best performance on real ICA data in all metrics on all three projection planes, compared to DeepCAv2 and DeepCA. For the first projection plane, IterCA performs better than DeepCAv2 and DeepCA with improvements of 1.18% and 3.61%, respectively in terms of $Ot(0)$ (same as Dice score). On the second and additional projection planes, which are mostly affected by motion, our IterCA method presents improvements of 15.22% and 6.78% in terms of $Ot(1)$ and 13.97% and 8.19% in terms of CD_{ℓ_2}, respectively, compared to DeepCA. This indicates our IterCA method has a better generalisation ability to the unseen real ICA data.

Table 6.3: Evaluation results of our proposed IterCA, DeepCAv2, and DeepCA on the unseen real ICA data for all three projection planes. Best results are annotated in **bold**.

Method	1^{st} Projection	2^{nd} Projection		Extra Projection	
	$Ot(0) \uparrow$	$Ot(1) \uparrow$	$CD_{\ell_2} \downarrow$	$Ot(1) \uparrow$	$CD_{\ell_2} \downarrow$
DeepCA [245]	$0.83_{\pm 0.04}$	$0.46_{\pm 0.07}$	$4.51_{\pm 1.29}$	$0.59_{\pm 0.04}$	$2.81_{\pm 0.06}$
DeepCAv2	$0.85_{\pm 0.04}$	$0.51_{\pm 0.08}$	$4.19_{\pm 1.26}$	$0.63_{\pm 0.08}$	$2.63_{\pm 0.94}$
IterCA	$\mathbf{0.86}_{\pm 0.04}$	$\mathbf{0.53}_{\pm 0.09}$	$\mathbf{3.88}_{\pm 1.36}$	$\mathbf{0.63}_{\pm 0.07}$	$\mathbf{2.58}_{\pm 0.88}$

All values represent mean ± standard deviation.

Statistical analysis

We use the ASO test [205, 206] as implemented by [207] to compare score distributions from different models, which is designed specifically for deep learning models. ASO returns a confidence score ϵ_{min}, which indicates (an upper bound to) the amount of violation of stochastic order. In our work, we set the rejection threshold $\tau = 0.25$ and we choose a significance level $\alpha = 0.05$. We calculate the confidence scores ϵ_{min} of ASO test for DeepCAv2 compared with DeepCA and IterCA compared with DeepCAv2 on CCTA, CCTA-UNSW, and VG datasets with different misalignment levels, in terms of $Ot(1)$ and CD_{ℓ_2}. The quantitative results of the ASO test are demonstrated in table 6.4.

For CD_{ℓ_2}, DeepCAv2 is tested to be stochastically dominant over DeepCA among all datasets, except for VG dataset with no and mild misalignment levels. Our IterCA is tested to be stochastically dominant over DeepCAv2 across all datasets with all different misalignment levels. In terms of $Ot(1)$, DeepCAv2 is tested to be stochastically dominant over DeepCA among 9 datasets out of 15. IterCA is tested to be stochastically dominant over DeepCAv2 among 7 datasets out of 12 and in the remaining 5 datasets, the confidence scores ϵ_{min} are 0.26, 0.27, 0.27, 0.30 and 0.38, which are close to the threshold $\tau = 0.25$. Therefore, it can be concluded that IterCA is tested to be stochastically dominant over DeepCAv2 regarding $Ot(1)$ across all scenarios overall.

We additionally present box plots for the evaluation results of our IterCA, DeepCAv2, and DeepCA on CCTA, CCTA-UNSW, and VG datasets simulated with different misalignment levels, in terms of $Ot(1)$ and CD_{ℓ_2}, as illustrated in figure 6.3 and figure 6.4. We can see

Table 6.4: Confidence scores ϵ_{min} of ASO test for DeepCAv2 compared with DeepCA and IterCA compared with DeepCAv2 on CCTA, CCTA-UNSW and VG datasets with different misalignment levels (namely, none, mild, medium, strong, and severe), in terms of $Ot(1)$ and CD_{ℓ_2}. Values are in **bold** if $\epsilon_{\text{min}} < \tau = 0.25$.

Misalignment Levels		None		Mild		Medium		Strong		Severe	
Dataset	**ASO (ϵ_{min})**	$Ot(1)$	CD_{ℓ_2}	$Ot(1)$	CD_{ℓ_2}	$Ot(1)$	CD_{ℓ_2}	$Ot(1)$	CD_{ℓ_2}	$Ot(1)$	CD_{ℓ_2}
CCTA	DeepCAv2 vs DeepCA	0.42	**<1e-4**	**0.01**	**<1e-4**	**0.02**	**<1e-4**	**0.01**	**<1e-4**	**0.01**	**<1e-4**
	IterCA vs DeepCAv2	N/A		0.38	**<1e-4**	0.27	**<1e-4**	**0.20**	**<1e-4**	0.30	**<1e-4**
CCTA - UNSW	DeepCAv2 vs DeepCA	0.38	**<1e-4**	**0.24**	**<1e-4**	**0.24**	**<1e-4**	**0.06**	**<1e-4**	**0.03**	**<1e-4**
	IterCA vs DeepCAv2	N/A		**0.23**	**<1e-4**	0.27	**<1e-4**	0.26	**<1e-4**	**0.16**	**<1e-4**
VG	DeepCAv2 vs DeepCA	0.85	0.83	0.50	0.68	0.35	**<1e-4**	**0.17**	**<1e-4**	0.50	**<1e-4**
	IterCA vs DeepCAv2	N/A		**0.20**	**<1e-4**	**0.08**	**<1e-4**	**0.03**	**<1e-4**	**0.01**	**<1e-4**

that IterCA has the overall best and most stable performance across all scenarios. The reason behind stable performance of IterCA across different misalignments is no matter how severe the misalignment is, IterCA can gradually mitigate the negative impact of large misalignments through iterative registration, which matches the purpose of our design. Therefore, this iterative process increases the model's elastic generalisation capacity.

6.3.3 Qualitative Results on CCTA Test Data

For each misalignment level, we display reconstruction results by our IterCA, DeepCAv2, and DeepCA on one CCTA test data combined with the corresponding ground truth in the same 3D space, as illustrated in figure 6.5. It illustrates that our IterCA can correct more deformation than DeepCAv2 and DeepCA, as IterCA presents better reconstruction overlap. It also shows that the performance of our IterCA method is more stable across different misalignments. More qualitative results on CCTA, CCTA-UNSW, and VG datasets are presented in appendix C.

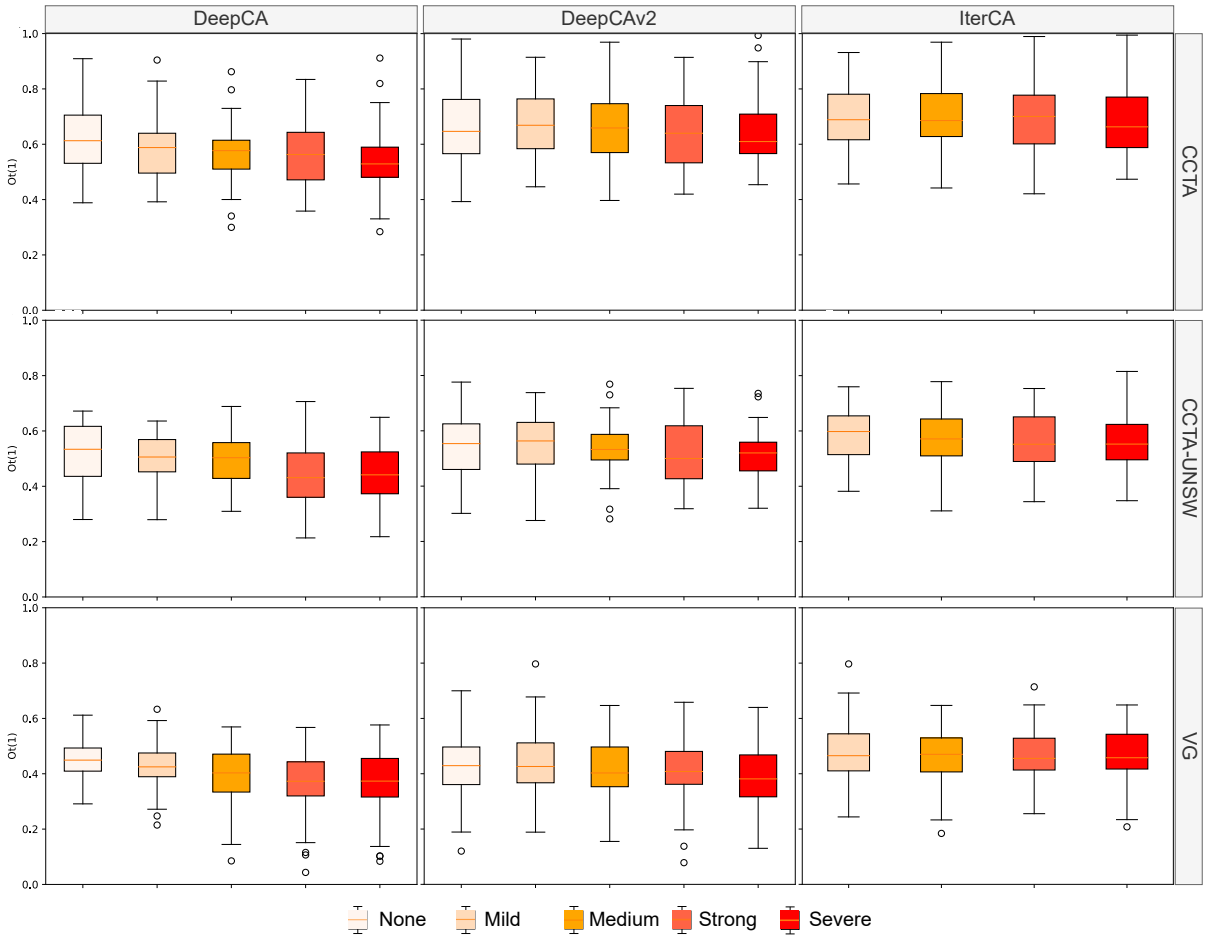


Figure 6.3: Box plots for the evaluation results of IterCA, DeepCAv2, and DeepCA (right to left) on CCTA, CCTA-UNSW, and VG datasets (top to bottom) with different misalignment levels (presented in different coloured boxes), regarding $Ot(1)$.

6.3.4 Qualitative Results on Clinical ICA Data

Figure 6.6 shows the 3D coronary tree reconstruction by IterCA, DeepCAv2, and DeepCA on three real ICA data. From the green box marked, we can see that our IterCA can generalise better to real ICA data, preserving vessel topology, maintaining branch connectivity and recovering missing branches.

Figure 6.7 illustrates two examples of qualitative results by our IterCA method on real ICA data for the first, second, and additional projections, which presents good vessel connectivity. In the second projection plane of patient P_2, the reconstructed posterior descending arteries are shorter as marked by the blue box. This may have been caused by the non-simultaneous ICA acquisitions where the contrast agent arrives at different times

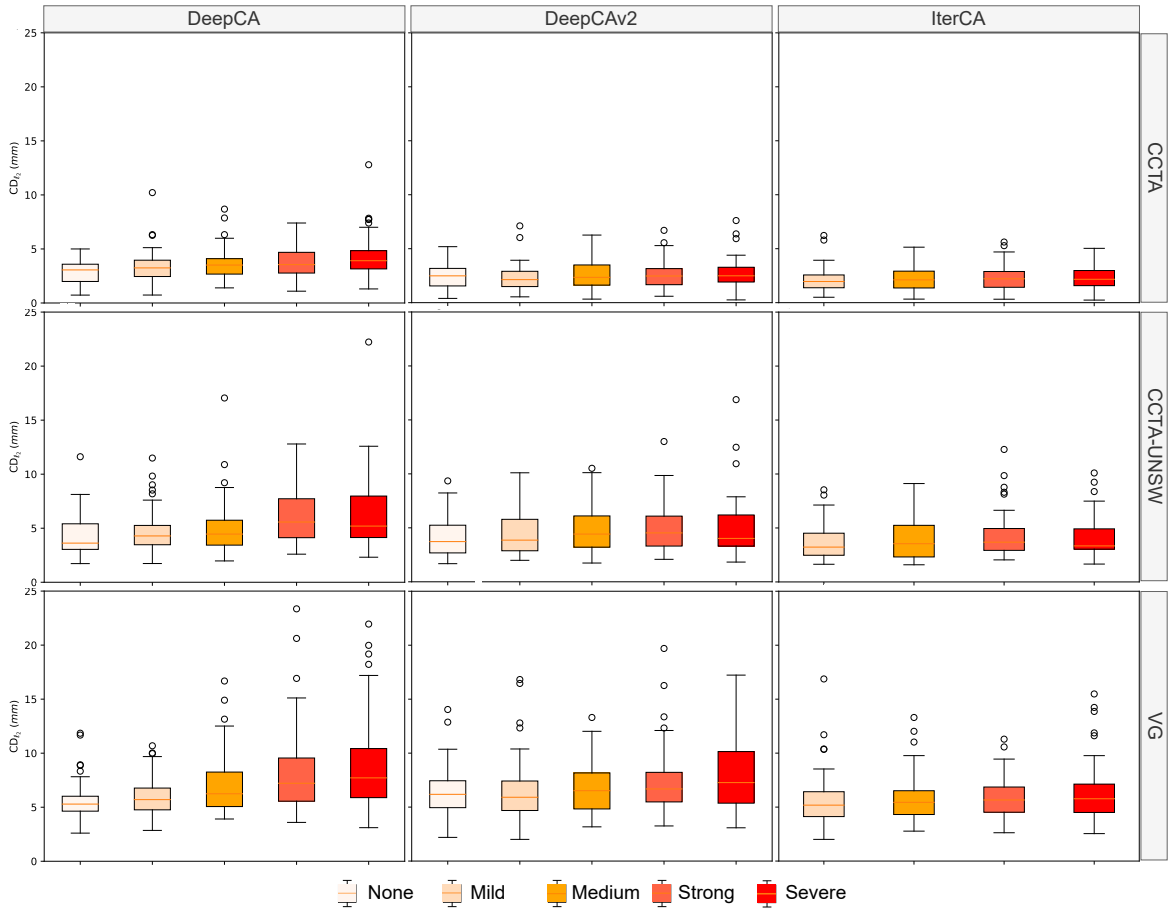


Figure 6.4: Box plots for the evaluation results of IterCA, DeepCAv2 and DeepCA (right to left) on CCTA, CCTA-UNSW and VG datasets (top to bottom) with different misalignment levels (presented in different coloured boxes), regarding CD_{ℓ_2} (mm).

between projections causing differences in the images. The qualitative comparison on the third projection plane in figure 6.7 also shows that even without involving the information of the additional projection plane in the input, our model can still accurately reconstruct the vascular structures in this plane.

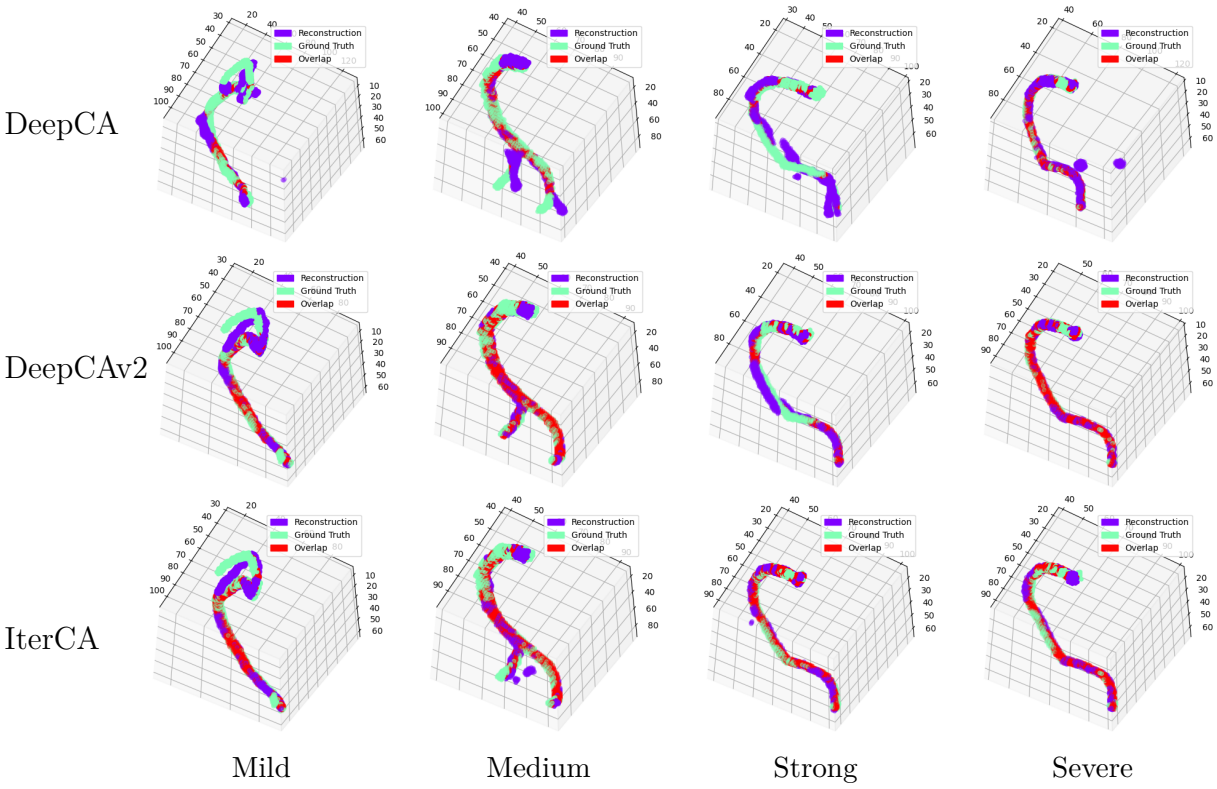


Figure 6.5: Reconstruction results by our IterCA, DeepCAv2, and DeepCA (bottom to top) on four different misalignment levels (left to right) of CCTA data combined with corresponding ground truth in the same 3D space. Colour purple represents the reconstruction, green the ground truth, and red the overlap.

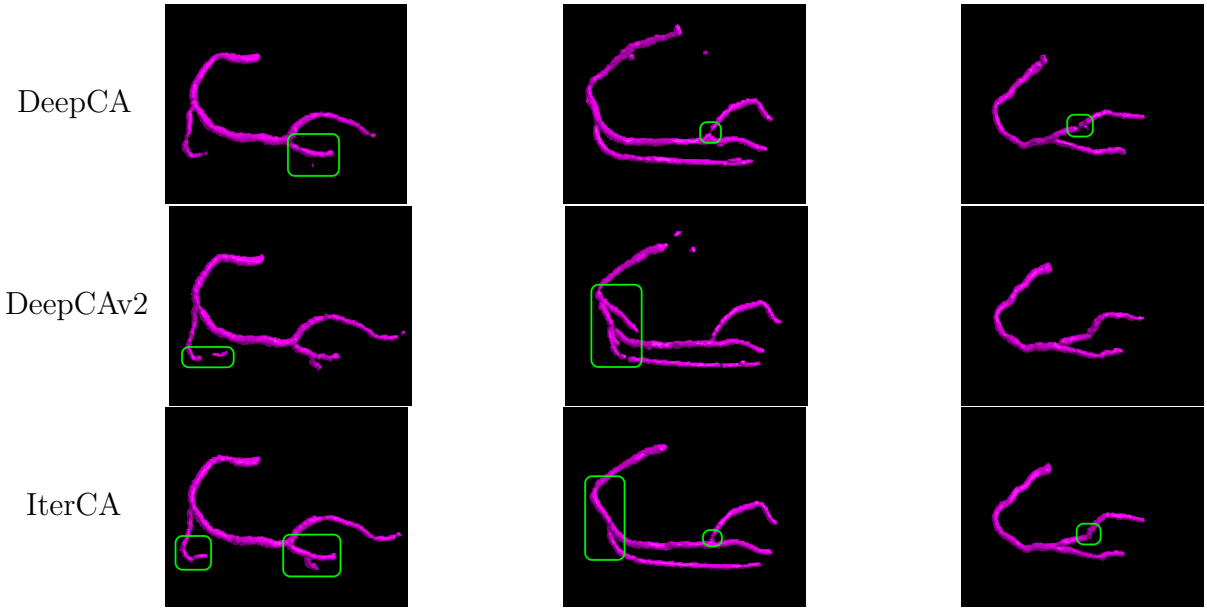


Figure 6.6: 3D coronary artery tree reconstruction results on three clinical ICA data (left to right) by our IterCA, DeepCAv2, and DeepCA models (bottom to top).

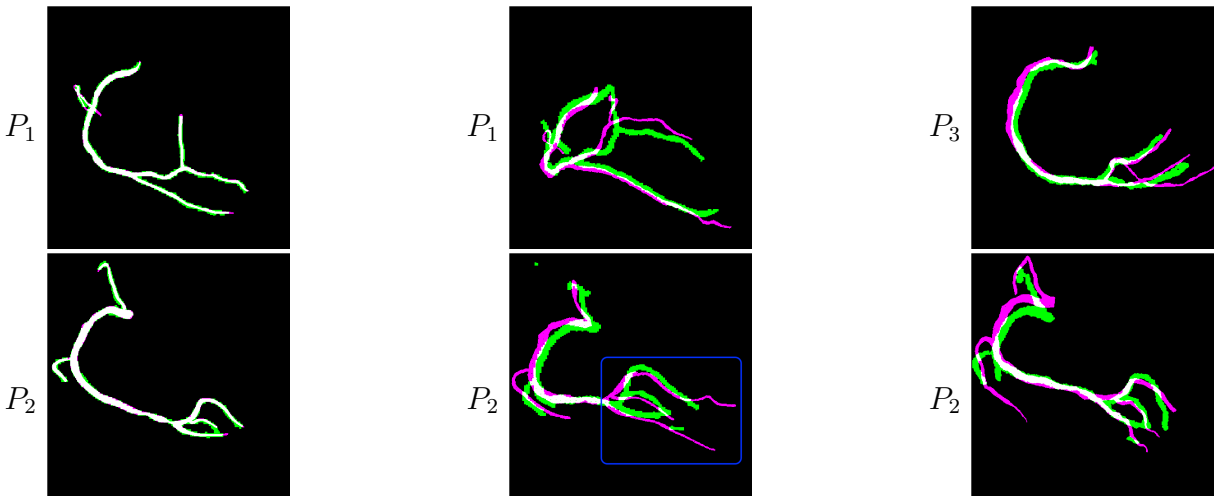


Figure 6.7: Two qualitative examples for each projection plane by our IterCA method. $P_{1,2,3}$ are different patients. From left to right: comparisons between the real ICA data and our reprojections on the first, second, and third projection planes after rigid registration. Colour purple represents ICA data, green represents reprojection, and white shows the overlap.

6.4 Discussion and Conclusion

The initial rigid registration is critical for the success of the following iterative steps and incorrect initial registration can directly lead to reconstruction refining failure. In our experiments, we tried typical intensity-based image rigid registration, which failed easily when there is large deformation between the reprojections and original ICA data. This sometimes resulted in wrong rotation of the ICA images to match the reprojection and got stuck in local optimum. Our IterCA utilised ICP at the start to reduce the initial large displacement, which performs better than the intensity-based registration. This is probably due to that our data are binary, so point-wise geometry-based registration is more suitable. ICP has the advantages of high accuracy and reliability in aligning two point clouds. In ICP, we leveraged distance thresholding for outlier rejection and convergence control, so it is less sensitive to outliers. In worst cases when the reprojections have many missing features, the overall structures can still be matched by ICP. This has been illustrated in figures B.5 and B.6 when some models generate 3D reconstruction with large missing features, the 2D reprojections of which can still be registered with the ICA data. However, due to point cloud representation, the ICA data may be sparse in

pixel-based image after registration. As shown in figure 6.7, we can see the registered ICA data (purple) on the second projection plane of P_2 are somewhat hollow, but this impacts less on our iterative refining process compared to false initial registration.

After initial registration, if we continue using rigid registration, there would be no more obvious alignment and so no more features can be gained from the 2D space to 3D. Hence, we leveraged non-rigid registration to further improve the contour matching. We use the registration in 2D space between the reprojections and ICA data as a compromising strategy to compensate for the motion in 3D, so this is not an exact motion correction, but an estimation.

Our quantitative evaluation on both CCTA test dataset and three unseen datasets demonstrates the superior performance of our IterCA on 3D coronary artery tree reconstruction from two non-simultaneous projections compared to both DeepCA and DeepCAv2. By simulating projections from CCTA data following real coronary angiography geometry, we achieve generalisation on real non-simultaneously acquired ICA data. The performance of our proposed different misalignments simulation demonstrates the effectiveness in generalisation to various real-world scenarios. The iterative motion compensation approach via registration on the non-simultaneous projection plane enables stable reconstruction under different conditions. Through iterative registration, IterCA can gradually eliminate the negative effect of large motion and align the two non-simultaneous projection planes. Therefore, our IterCA shows better generalisation capability and stable performance across all scenarios as illustrated in figures 6.3 to 6.5, compared to DeepCA which could not steadily perform reconstruction, in particular, if there is motion out of simulated deformations.

The robustness of IterCA highlights its potential for clinical use, in helping clinicians to understand complex 3D coronary tree structures during cardiac interventions. Our IterCA method under 15 iterations can automatically reconstruct 3D coronary tree in nearly real time (approximately 10 seconds per iteration) compared to traditional methods, but requires pre-segmentation on ICA data. Although there are several methods for automated

coronary vessel segmentation [238, 239], the impact of automated segmentation quality on reconstruction performance should be further explored. In summary, we propose a novel iterative pipeline, IterCA, which provides a solid baseline for quantitative comparison of future work in the area of 3D coronary artery tree reconstruction from two clinical non-simultaneous angiographic projections.

Chapter 7

Snowflake Point Transformer

Abstract - PointCA: Snowflake Point Transformer with Point Adversarial Loss for Coronary Tree Reconstruction from Two Non-simultaneous Projections

The current best-performing deep learning-based methods for 3D coronary artery tree reconstruction from two non-simultaneous projections, such as DeepCA in chapter 5 and IterCA in chapter 6, rely on voxel-based representations. This makes them inefficient for the reconstruction of sparse, complex curvilinear structures such as coronary vessel trees, and hard to perform high-resolution reconstruction under limited computational resources. Alternative representations such as point clouds would overcome this limitation. In this chapter, a novel point cloud-based method named PointCA is proposed to efficiently reconstruct dense point cloud-based coronary artery tree surface from two non-simultaneous ICA projections. A novel network, Snowflake Point Transformer, with an expanded receptive field for efficient feature extraction and effective 3D generation of structural characteristics, and a novel Wasserstein conditional point adversarial learning module with gradient penalty designated for point cloud structures, are introduced. The PointCA method dynamically adapts to different point cloud sizes in the same input batch to support the different complexities of vessel structures. Through simulating projections from the CCTA data, the non-rigid motion between non-simultaneous projections is implicitly compensated. An application-specific evaluation metric is incorporated to validate the proposed PointCA method on both a public CCTA dataset with real patients' vessel anatomies and a real ICA dataset, together with Chamfer ℓ_2 distance. The results demonstrate the promising performance of the PointCA method in dense point cloud-based coronary artery tree surface reconstruction while largely reducing computational inefficiency.

7.1 Introduction

Deep learning methods for 3D coronary artery tree reconstruction can be categorised into five classes according to their reconstruction representations: voxel-based, tubular shape-based with centreline and radii, mesh-based, implicit neural-based, and Gaussian-based representations. Voxel representation-based deep learning techniques [168, 245] are, in general, inefficient due to the sparse structure of vessels in 3D. In [169–171], a tubular structure including vessel centreline and radii is predicted for the coronary tree reconstruction, but these methods are based on synthetic data that do not contain real patients' anatomies; they also require more than two projections for training. Bransby *et al.* [172] proposed mesh reconstruction from bi-planar data that can only reconstruct a single branch each time. Based on implicit neural representation, Maas *et al.* [173, 174] introduced a technique that requires at least 4 projections, and Zhu *et al.* [177] leveraged ICA frames from videos for dynamic cardiovascular reconstruction. The self-supervised NeCA model introduced by Wang *et al.* [240] needs around one hour to obtain a good reconstruction result. Fu *et al.* [178] incorporated a Gaussian representation for 3D reconstruction, without considering real clinical settings. Also, they require more than half an hour for reconstruction, and the results based on two projections are extremely coarse. Overall, most aforementioned methods have typically used synthetic data, CCTA data, ICA data from bi-planar scans, or ICA frames gated at the same cardiac phase assuming no residual temporal motion, none of which suffer from non-rigid motion between projections [168–174, 177, 178, 240]. This limitation makes previous methods ill-suited for real non-simultaneous ICA acquisitions. DeepCA [245] in chapter 5 and IterCA in chapter 6, as far as we have known, are the currently only deep learning-based approaches for 3D coronary artery tree reconstruction from two non-simultaneous projections. However, they rely on voxel-based representations, which are inefficient for the reconstruction of sparse, complex curvilinear structures, such as coronary vessel trees. It is also hard to perform high-resolution voxel-based reconstruction due to the limit of computational resources. Despite the improvement in deep neural networks, efficient 3D reconstruction of

a high-resolution coronary artery tree from two non-simultaneous angiographic projections has remained an open problem.

In this chapter, we propose a novel point cloud-based method, named PointCA, to efficiently reconstruct dense coronary artery trees from two 2D non-simultaneous projections. We introduce a novel network, Snowflake Point Transformer. We leverage an expanded receptive field to encode features by an efficient serialised neighbour mapping of point clouds organised with specific patterns. Based on the extracted features, we perform effective point deconvolution to generate locally compact and structured dense point clouds to reconstruct the 3D coronary artery tree shape characteristics with detailed geometries. We specifically propose a novel Wasserstein conditional point adversarial learning module with gradient penalty, designated for point cloud structures, allowing accurate distinguishing between point cloud reconstruction and ground truth. Our method enables a dynamic number of points from different point clouds in the same batch to support different complexities of vessel structures. To resemble real non-simultaneous ICA projections, we simulate 2D projections in different planes from the CCTA data containing real coronary tree geometries, with a rigid transformation simulation before forward projection. We then use these simulated projections to learn from the original high-resolution dense CCTA ground truth to enable generalisation to real non-simultaneous ICA projections. In this way, we not only implicitly compensate for the non-rigid motion between real clinical non-simultaneous projections, but also overcome the problems of both the limited number of real paired ICA data with projection geometry information and the unavailable 3D ground truth for real ICA data. We focus on the RCA in this study, because RCA undergoes more compressive strain and is affected more by motion artifacts than other coronary vessels. We provide an application-specific evaluation method to tackle the deformation in 3D reconstructions, the unavailability of 3D ground truth for real ICA scans, and the motion between projection planes, together with Chamfer ℓ_2 distance. We validate our proposed PointCA on a public CCTA dataset with real patients' vessel anatomies and a real ICA dataset (unseen domain), in comparison to three other models.

The results demonstrate the promising performance of our PointCA method in dense point cloud-based coronary artery tree reconstruction while largely reducing computational inefficiency. Our main contributions are:

1. **Efficient dense coronary artery tree reconstruction:** We leverage point cloud-based representation to efficiently handle vessel topology to achieve dense point cloud-based coronary artery tree reconstruction.

2. **Reconstruction from two clinical non-simultaneous projections:** Through simulating projections from the CCTA data, the non-rigid motion between projections is implicitly compensated to facilitate reconstruction from two non-simultaneous projections.

3. **Dynamic input:** Our method supports a dynamic size of different point clouds in one batch, allowing various vasculature reconstructions.

4. **Point adversarial loss:** Our proposed Wasserstein conditional point adversarial learning module with gradient penalty is specifically designed for point cloud structures, accurately discriminating between point cloud reconstruction and ground truth.

7.2 Methods

Our proposed PointCA method consists of two modules: a conditional generator and a conditional point adversarial learning module, as illustrated in figure 7.1. The conditional generator is based on a powerful novel snowflake point Transformer (SPT) module with the ability to dynamically support various point cloud sizes (different complexity of vessel structures) in one batch during training. It generates a corresponding coronary artery tree based on the input condition. The conditional point adversarial learning module leverages PCN encoders for effective feature extraction from point clouds. The encoded features from the generated results and the corresponding ground truth are concatenated with the

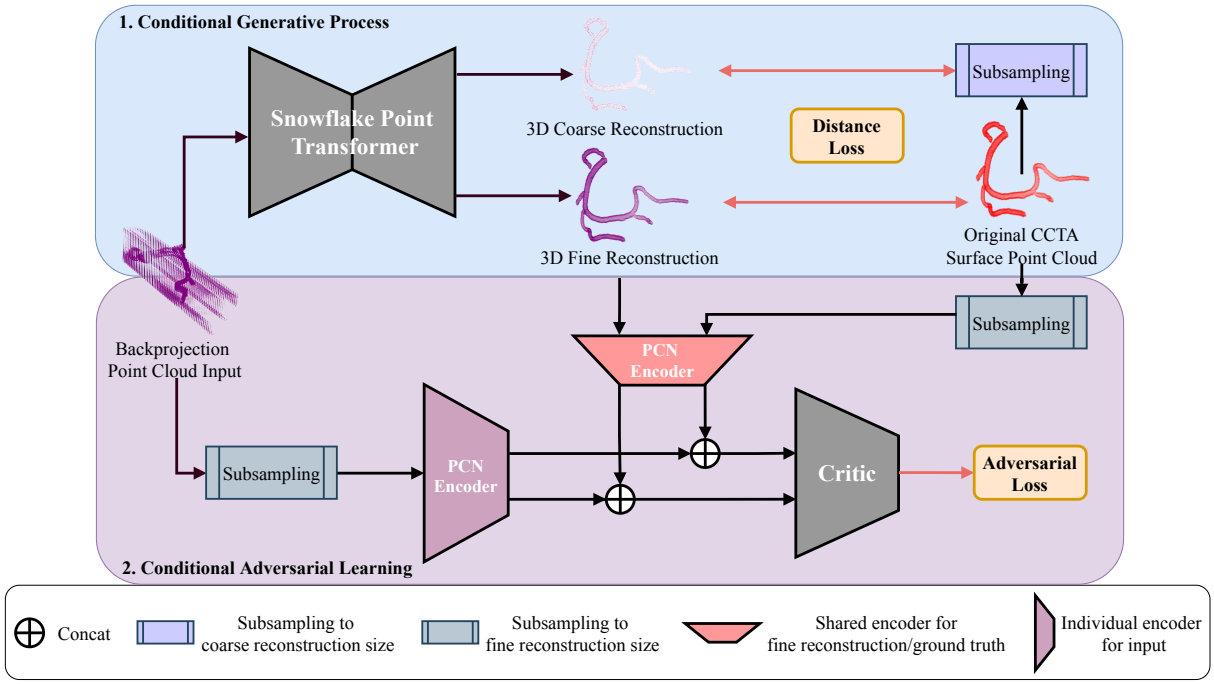


Figure 7.1: Framework of our proposed PointCA method, consisting of two modules: a conditional generator and a conditional point adversarial learning module. The conditional generator is based on the Snowflake Point Transformer that generates a corresponding coronary artery tree based on the input condition. The conditional point adversarial learning module contains PCN encoders for effective point cloud feature extraction. The encoded features from the generated results and the corresponding ground truth are concatenated with the encoded input features separately, which are then sent to the critic for adversarial training.

encoded input features separately, which are then sent to the critic that uses Wasserstein loss with gradient penalty for stable adversarial training.

7.2.1 Snowflake Point Transformer (SPT)

Our proposed SPT model leverages the scaling encoding from Point Transformer v3 [73] and the local detailed geometry generation ability from SPD [75]. It dynamically supports various point cloud sizes in one batch during training. The SPT includes four stages as described in the following subsections: **(a)** we present the construction design of dynamic point cloud input, **(b)** we introduce the input initialisation via point cloud serialisation and embedding, **(c)** we illustrate the design of our encoder layer consisting of patching, serialised pooling, and serialised transformer blocks, and **(d)** finally with the encoded features, we produce an initial coarse reconstruction via seed generator and the final fine

reconstruction via point generator based on multiple SPD layers.

Dynamic Point Cloud Input

Most point cloud-based models require the same number of points for each sample in the same batch during training. This sacrifices data quality as some simple structures would add redundant points and some complex structures may drop important feature points. We use a special batch handling approach conceptualised from PyTorch Geometric [256] to enable input point clouds of various sizes, as shown in figure 7.2.

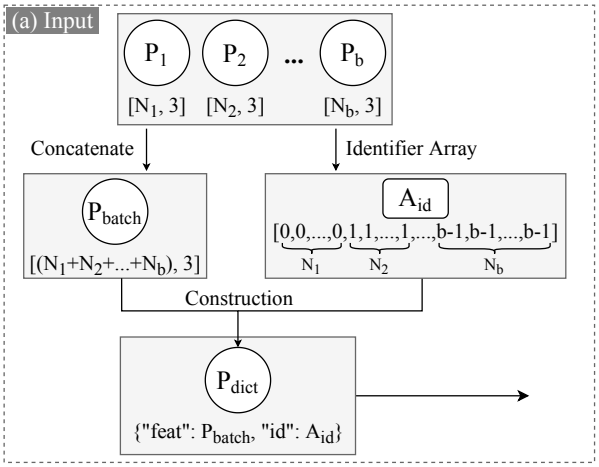


Figure 7.2: Input construction design of dynamic point cloud.

Let us assume we have B point clouds in one batch. Each point cloud $\mathcal{P}_b \in \mathbb{R}^{N_b \times C}$ contains N_b points with C features; here $C = 3$ for x, y, and z coordinates. We concatenate these B point clouds to obtain batched point cloud data $\mathcal{P}_{\text{batch}} \in \mathbb{R}^{(N_1+N_2+\cdots+N_B)\times C}$. In this way, multiple point clouds with different sizes can be trained at the same time. To distinguish points from different point clouds, we create an identifier array $\mathcal{A}_{\text{id}}$ shown in equation (7.1), where the number $b-1$ identifies a point cloud $\mathcal{P}_b$ and is repeated N_b times. The batched data $\mathcal{P}_{\text{batch}}$ and corresponding identifier array $\mathcal{A}_{\text{id}}$ are stored in a dictionary structure $\mathcal{P}_{\text{dict}} = \{\text{“feat”: } \mathcal{P}_{\text{batch}}, \text{“id”: } \mathcal{A}_{\text{id}}\}$, which is supported by our encoder.

$$\mathcal{A}_{\text{id}} = [\underbrace{0, 0, \cdots, 0}_{N_1}, \underbrace{1, 1, \cdots, 1}_{N_2}, \cdots, \underbrace{B-1, B-1, \cdots, B-1}_{N_B}]. \tag{7.1}$$

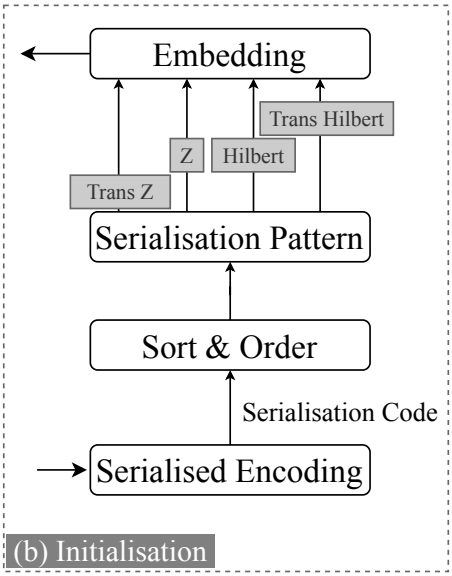


Figure 7.3: The whole point cloud initialisation process.

Point Cloud Input Initialisation

We use the point cloud serialisation strategy from [73] that transforms unstructured point clouds into a structured format, as illustrated in figure 7.3. The transformation includes four types of space-filling curves, namely 'Z-order', 'Hilbert', 'Trans Z-order', and 'Trans Hilbert' curves, that are specific design paths connecting every point. These different curve shapes offer various perspectives on spatial relationships, extracting many possible particular features. We employ serialised encoding [73] to encode each point's position to a serialisation code, representing its order within a specific curve. We then sort points based on the code to make the batched point cloud data ordered with the selected curve pattern. In this way, neighbouring points tend to be near in space. After serialisation, we perform sparse convolution (SpConv) [257] on each serialised point cloud to get the embedding features $\mathcal{P}_{\text{embed}}$:

$$\mathcal{P}_{\text{embed}} = \text{SpConv}(\text{Ordering}_{(4 \text{ curve patterns})}(\text{SerialisedEncoding}(\mathcal{P}_{\text{dict}}))). \tag{7.2}$$

Serialised Point Transformer Encoder

Our encoder layer contains a serialised pooling layer and S_{block} serialised transformer blocks, as illustrated in figure 7.4. Before serialised pooling, we first perform patch

grouping to group points into non-overlapping patches and then apply patch interaction ($\mathcal{P}_{\text{patches}} = \text{Patching}(\mathcal{P}_{\text{embed}})$). We reorder the point cloud based on the order derived from a specific serialisation pattern. Next, we pad the point cloud sequence by borrowing points from neighbouring patches to ensure it is divisible by the designated patch size S_{patch}. Patch interaction is applied between points from different patches $\mathcal{P}_{\text{patches}}$ to integrate information across the entire point cloud, overcoming in this way the limitations of a non-overlapping architecture and enabling later patch attention to be effective. We use the Shuffle Order [73] strategy for patch interaction, where the permutations of serialised orders are randomised.

$$\mathcal{P}_{\text{patches}} = \text{Patching}(\mathcal{P}_{\text{embed}}) = \text{ShuffleOrder}(\text{Grouping}_{S_{\text{patch}}}(\mathcal{P}_{\text{embed}})). \tag{7.3}$$

After patch grouping and interaction, we apply serialised pooling (Pooling) following the grid pooling introduced in [258] to value non-overlapping receptive fields and fuse points within each non-overlapping grid cell, which is fast and has good generalisability. The serialised transformer block contains positional encoding (PosEncoding), serialised attention (Atten), and an MLP. The positional encoding is realised by a SpConv layer with a skip connection and the serialised attention [73] contains $S_{\text{attenHead}}$ heads. We next perform serialised attention [73] with $S_{\text{attenHead}}$ heads and MLP for each individual patch with a pre-norm structure by Layer Normalisation (LN). The previous shifting interaction across the four serialisation patterns enables various receptive fields for each attention layer, so it can functionally broaden the attention's receptive field by increasing the patch size S_{patch} while still preserving spatial neighbour relationships to a feasible extent.

We perform S_{enc}-level encoder layers to obtain our latent shape code $\mathbf{f} \in \mathbb{R}^{B \times C'}$ as shown in equation (7.4), where $S_{\text{enc}} = 4$ and $\leftarrow$ mean concatenation. For the 4 encoder layers, we set $S_{\text{block}} = [2, 2, 2, 2]$, $S_{\text{attenHead}} = [4, 8, 16, 32]$, and $S_{\text{patch}} = [128, 128, 128, 128]$. Through initial embedding and serialised pooling in encoder layers, we increase the feature channels

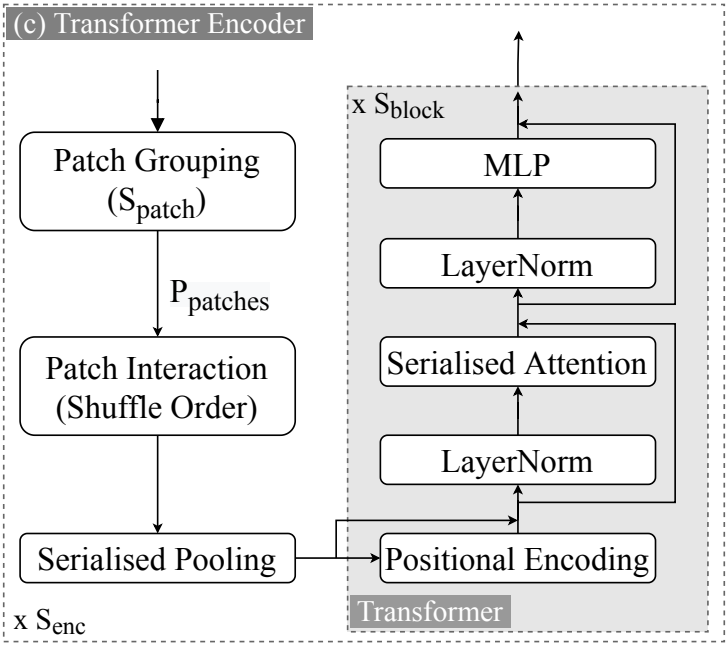


Figure 7.4: Structure of serialised point transformer encoder.

to $C' = 256$.

$$\mathbf{f} = \underbrace{\underbrace{\overbrace{\mathrm{MLP}(\mathrm{LN}(}^{}\overbrace{\mathrm{Atten}(\mathrm{LN}(}^{}\overbrace{\mathrm{PosEncoding}(}^{}\mathrm{Pooling}(\mathrm{Patching}(\mathcal{P}_{\mathrm{embed}}))))))}_{\text{Transformer blocks: } S_{\mathrm{block}} \text{ times}}}_{\text{Encoder layers: } S_{\mathrm{enc}} \text{ times}} . \tag{7.4}$$

Snowflake Point Reconstruction Decoder

The decoder contains two modules: seed generator and point generator, as illustrated in figure 7.5. Based on the shape code $\mathbf{f}$ generated from our encoder, we use the seed generator to produce coarse but complete reconstructed point clouds $\mathcal{R}_0 \in \mathbb{R}^{B \times M_0 \times 3}$ to capture the geometry and structure of the underlying shape, where $M_0 = 2,048$ is the coarse number of points. The seed generator is built on a point-wise splitting operation [75] following two MLPs. The point-wise splitting operation produces several child points for the input point cloud, which is based on one-dimensional deconvolution. $\mathcal{R}_0$ then serves as the seed coarse point cloud to the next module.

The point generator is based on S_{dec} blocks of SPD [75], aiming to generate more points and reconstruct the detailed local geometric pattern. SPD adds variations (more points) that

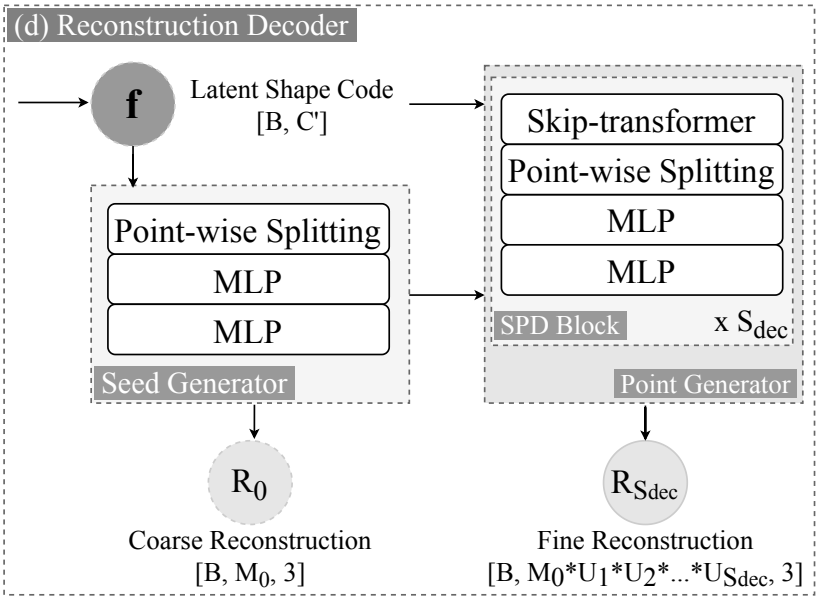


Figure 7.5: Structure of snowflake point reconstruction decoder.

comply with local patterns to the parent features via a point-wise splitting operation. SPD takes point clouds from the previous step together with the shape code $\mathbf{f}$ and splits each point cloud to expand the number of points by upsampling factors ($U = [U_1, U_2, \cdots, U_{S_{\text{dec}}}]$) to get new refined reconstructed point clouds ($\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_{S_{\text{dec}}} \in \mathbb{R}^{B\times(M_0*U_1*U_2*\cdots*U_{S_{\text{dec}}})\times 3}$), where $\mathcal{R}_{S_{\text{dec}}}$ is the final fine point cloud reconstruction output. We set $S_{\text{dec}} = 2$ and $U = [3, 3]$, so we finally generate every fine point cloud with dense size $M_0 \times U_1 \times U_2 = 2048 \times 3 \times 3 = 18,432$. A skip-transformer [75] is used in SPD to enable coherent collaboration between SPD blocks, which can remember the historic splitting information to adjust the splitting pattern in consecutive SPD blocks, keeping the pattern of local patches from overlapping with each other.

7.2.2 Conditional Point Adversarial Learning

We propose a Wasserstein conditional point adversarial learning module with gradient penalty that specifically supports point cloud structures with the incorporation of a PCN encoder for critical point cloud feature extraction. The conditional structure [60] enables us to discriminate generative results given a specific input condition, and the Wasserstein adversarial objective with an additional gradient penalty constraint [59] improves training

stability.

After the generative process by the SPT module, we obtain the predicted fine reconstruction results $\mathcal{R}_{S_{\text{dec}}}$. Let us assume the collection of all input point clouds of various sizes in one batch as $\mathcal{P}_{\text{input}}$ and the corresponding ground truth collection as $\mathcal{P}_{\text{gt}}$. In order to employ the PCN encoder, we first subsample each point cloud from both $\mathcal{P}_{\text{input}}$ and $\mathcal{P}_{\text{gt}}$ to the same point cloud size as the fine reconstruction. We adopt the point cloud subsampling strategy from PointNet++ [72]. We then use one shared PCN encoder to encode both subsampled $\mathcal{P}_{\text{gt}}$ and fine reconstruction results $\mathcal{R}_{S_{\text{dec}}}$ to obtain the encoded features $\mathbf{y}$ and $\widehat{\mathbf{y}}$, respectively, and an additional individual encoder to encode subsampled $\mathcal{P}_{\text{input}}$ to get the extracted features $\mathbf{x}$. Next, the encoded features $\widehat{\mathbf{y}}$ and $\mathbf{y}$ from the generated results and the corresponding ground truth are concatenated ($\oplus$) with the encoded conditional input features $\mathbf{x}$ separately, which are sent to the critic D, an MLP-based network.

$$\widehat{\mathbf{y}}_{\mathbf{x}} = \widehat{\mathbf{y}} \oplus \mathbf{x}, \quad \mathbf{y}_{\mathbf{x}} = \mathbf{y} \oplus \mathbf{x}. \tag{7.5}$$

The critic D leverages Wasserstein loss with gradient penalty for stable adversarial training. Specifically, $\widehat{\mathbf{y}}_{\mathbf{x}}$ and $\mathbf{y}_{\mathbf{x}}$ are used in the critic D to approximate the Wasserstein distance (or, Earth-Mover distance) $W((\mathbb{P}_r)_{\mathbf{y}_{\mathbf{x}} \sim \mathbb{P}_r}, (\mathbb{P}_g)_{\widehat{\mathbf{y}}_{\mathbf{x}} \sim \mathbb{P}_g})$ and gradient penalty constraint $GP(\mathbb{P}_{\mathbf{y}_{\mathbf{x}}})$ for each data batch, where $\mathbb{P}_r$ is the conditional ground truth data distribution, $\mathbb{P}_g$ is the conditional generative fine reconstruction distribution, and $\mathbb{P}_{\mathbf{y}_{\mathbf{x}}}$ is the distribution sampling uniformly along straight lines between pairs of points sampled from the distributions $\mathbb{P}_r$ and $\mathbb{P}_g$.

$$W((\mathbb{P}_r)_{\mathbf{y}_{\mathbf{x}} \sim \mathbb{P}_r}, (\mathbb{P}_g)_{\widehat{\mathbf{y}}_{\mathbf{x}} \sim \mathbb{P}_g}) = \mathop{\mathbb{E}}_{\mathbf{y}_{\mathbf{x}} \sim \mathbb{P}_r}[D(\mathbf{y}_{\mathbf{x}})] - \mathop{\mathbb{E}}_{\widehat{\mathbf{y}}_{\mathbf{x}} \sim \mathbb{P}_g}[D(\widehat{\mathbf{y}}_{\mathbf{x}})]. \tag{7.6}$$

$$GP(\mathbb{P}_{\mathbf{y}_{\mathbf{x}}}) = \mathop{\mathbb{E}}_{\mathbf{y}_{\mathbf{x}} \sim \mathbb{P}_{\mathbf{y}_{\mathbf{x}}}}[(\|\nabla_{\mathbf{y}_{\mathbf{x}}} D(\mathbf{y}_{\mathbf{x}})\|_2 - 1)^2], \text{where } \mathbf{y}_{\mathbf{x}} = \epsilon \mathbf{y}_{\mathbf{x}} + (1-\epsilon)\widehat{\mathbf{y}}_{\mathbf{x}}, \quad \epsilon \in U[0,1]. \tag{7.7}$$

7.2.3 Loss Function

During training, the SPT generator G tries to minimise $W(\mathbb{P}_r, \mathbb{P}_g)$ between distributions $\mathbb{P}_r$ and $\mathbb{P}_g$, while the critic D tries to maximise this distance along with minimising the

constraint $GP(\mathbb{P}_{\mathbf{y_x}})$. The objective function of the adversarial learning is presented in equation (7.8), where λ_1 is the penalty coefficient. We use $\lambda_1 = 10$.

$$\begin{aligned}\mathcal{L}_{\text{adversarial}}(G, D) = &\arg\min_{G}\max_{D} W((\mathbb{P}_r)_{\mathbf{y_x}\sim\mathbb{P}_r}, (\mathbb{P}_g)_{\hat{\mathbf{y}}_\mathbf{x}\sim\mathbb{P}_g}) \\ &+ \lambda_1 \min_{D} GP(\mathbb{P}_{\mathbf{y_x}}).\end{aligned} \tag{7.8}$$

We additionally impose both CD_{ℓ_2} and *EMD* as the distance reconstruction loss function between the ground truth and our reconstruction results. *EMD* is defined based on two point sets which have the same size; we follow the *EMD* implementation from [204]. Our reconstruction loss $\mathcal{L}_{\text{recon}}$ is defined as:

$$\begin{aligned}\mathcal{L}_{\text{recon}} &= \mathcal{L}_{\text{coarse}}(\text{EMD}) + \mathcal{L}_{\text{coarse}}(\text{CD}_{\ell_2}) + \lambda_2 \mathcal{L}_{\text{fine}}(\text{CD}_{\ell_2}) \\ &= \mathcal{L}_{\text{EMD}}(\mathcal{R}_0, \mathcal{P}'_{\text{gt}}) + \mathcal{L}_{\text{CD}_{\ell_2}}(\mathcal{R}_0, \mathcal{P}'_{\text{gt}}) + \lambda_2 \mathcal{L}_{\text{CD}_{\ell_2}}(\mathcal{R}_{S_{\text{dec}}}, \mathcal{P}_{\text{gt}}),\end{aligned} \tag{7.9}$$

where λ_2 is a weight to control losses between coarse and fine reconstruction, $\mathcal{R}_0$ the coarse reconstruction, $\mathcal{R}_{S_{\text{dec}}}$ the fine reconstruction, $\mathcal{P}_{\text{gt}}$ the original ground truth point cloud, and $\mathcal{P}'_{\text{gt}}$ denotes the downsampled ground truth matching the size of coarse reconstruction $\mathcal{R}_0$. We set $\lambda_2 = 0.5$ at the start to emphasise the coarse reconstruction initially and increase its value to 1 as training progresses.

Our final objective function $\mathcal{L}$ consists of both adversarial objective $\mathcal{L}_{\text{adversarial}}(G, D)$ and reconstruction loss $\mathcal{L}_{\text{recon}}$, defined in equation (7.10):

$$\mathcal{L} = \mathcal{L}_{\text{adversarial}}(G, D) + \lambda \mathcal{L}_{\text{recon}}. \tag{7.10}$$

The hyperparameter λ is set to 100 after fine-tuning. The number of critic iterations per generator iteration n_{critic} is set to 1.

7.3 Experimental Settings

7.3.1 Datasets and Data Preprocessing

We use a public CCTA dataset [222] containing 3D RCA anatomy of real patients, where we use 810 RCA anatomies in total, dividing them into 75% training, 15% validation, and

10% test datasets. We also collect a clinical ICA dataset of 8 patients for evaluation, who were admitted at the Oxford John Radcliffe Hospital with suspected coronary stenosis and provided informed consent. For two of the patients, three ICA projections were captured, while for the rest, there were two ICA projections.

Data Preprocessing

We follow the projection geometry from real clinical scenarios to simulate the two cone-beam forward projections from the original CCTA data using the TIGRE toolbox [220]. Details of the projection geometry parameters are the same as table 5.1 in chapter 5, with the only difference that we follow the original resolution (volume size and voxel spacing) of each CCTA data. In real clinical scenarios, breathing and cardiac motions introduce deformations to the coronary artery tree between projections. In order to implicitly compensate for these motion artifacts between non-simultaneous projections by our PointCA model, in the generation of two simulated projections, we introduce rigid transformations to the CCTA data before performing forward projection on the second projection plane to simulate motion. We follow the medium misalignment level proposed in table 6.1 of chapter 6 for simulating rigid transformations, but we sample translation and rotation parameters from a uniform distribution instead of a normal distribution, since this work does not iteratively reduce motion between projections for reconstruction. Two projections generated from each 3D CCTA data are backprojected into 3D voxel space using the known projection geometry. Voxels where any backprojected line crosses form an initial mask. We then use coordinates of the voxels on the surface to build an initial backprojection surface point cloud input that contains non-aligned projection information, via the Euclidean Distance Transform. We perform an equivalent surface extraction on the original high-resolution CCTA volumes to get the dense point cloud ground truth. According to the input point cloud size distribution as illustrated in figure 7.6, we subsample any input point cloud with size $> 40,000$ to size $40,000$ for efficiency, while keeping the rest of the original varied sizes. The clinical ICA data

are pre-segmented and then backprojected to form the surface point cloud input before evaluation.

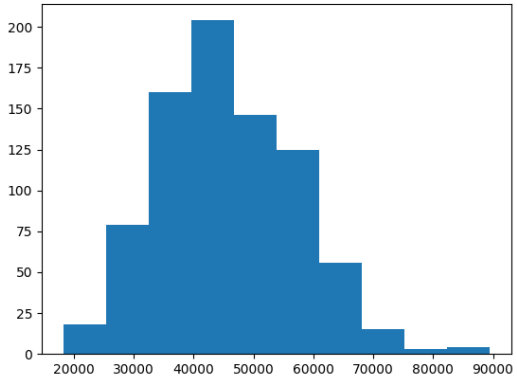


Figure 7.6: The input point cloud size distribution. Vertical axis stands for the number of point clouds and horizontal axis presents the point cloud size.

7.3.2 Baseline Models and Implementation Details

For comparative analyses, we implement three point cloud-based baseline models, including the SOTA point cloud reconstruction model SnowflakeNet [75] and the traditional model PCN [74]. The third baseline is based on the latest point cloud-based network architecture, Point Transformer v3 [73] with replacement of its decoder with SnowflakeNet decoder to enable coarse-to-fine reconstruction, so it is the same as model SPT. All three baseline models generate an initial coarse point cloud reconstruction with size $2,048$ and a final fine point cloud reconstruction with size $18,432$, same as our PointCA method. All three baselines also adopt the reconstruction training loss $\mathcal{L}_{\text{recon}}$ which is described in equation (7.9), the same as our PointCA method without adversarial loss. For SnowflakeNet and PCN that do not support input point clouds with dynamic sizes, we subsample all input point clouds to size $40,000$ when training these two models. We keep the original varied sizes of the input point clouds during testing for all the models, so in terms of SnowflakeNet and PCN, we evaluate each test data individually instead of in batches.

We use AdamW optimiser [259] with weight decay of 0.01. The initial learning rate is set

to 5e-4. For the serialised transformer blocks in model SPT, the initial learning rate is individually set to 5e-5. For our conditional adversarial learning module, we use Adam optimiser [236], with an initial learning rate of 10e-4. We use a warmup strategy for the first 24 epochs out of a total of 600 epochs and then adopt a cosine annealing strategy. Training is performed with a batch size of 15 on an HPC cluster utilising Nvidia Tesla v100 GPUs.

7.3.3 Metrics

We adopt the overlap using a sweeping distance threshold ($Ot(d)$) as an evaluation metric, where d is the distance threshold in mm unit [202]. $Ot(d) \in [0, 1]$ with 0 representing no overlap and 1 the perfect match. The different d values allow us to measure point cloud reconstructions under different degrees of deformation. In addition, we use the CD_{ℓ_2} for measuring the corresponding point-wise or pixel-wise prediction errors (mm) in either 3D point clouds or 2D projection data according to their point or pixel spacing. We also measure the similarity between our reconstructed point cloud and the ground truth using EMD, which is more discriminative to the density distribution, compared to CD_{ℓ_2}.

We evaluate the models on both the CCTA test dataset and the unseen real clinical ICA dataset. For the CCTA test dataset, we directly validate the point cloud results in 3D space after rigidly registering the ground truth to the predicted reconstruction using $Ot(d)$ with $d = \{1, 2\}$ mm, CD_{ℓ_2}, and EMD. In terms of the real clinical ICA dataset, we evaluate the performance using 2D reprojection of the predicted reconstructions by different models, as we only have 2D ICA data instead of 3D ground truth. Since point clouds are not confined to fixed positions like voxel grids, there exists deformation on all 2D projection planes. For this reason, when measuring the clinical results on any 2D projection planes, we first rigidly register the original ICA data to the reprojections. We then compute the $Ot(d)$ with $d = \{1, 2\}$ mm and CD_{ℓ_2} between them. All the rigid registration is based on the ICP algorithm [251, 252]. All the 2D reprojections are binarised with a threshold of 0 before evaluation and visualisation.

Table 7.1: Quantitative performance on point cloud CCTA test dataset of our proposed PointCA and three baseline models, namely PCN, SnowflakeNet, and SPT, in terms of four metrics, i.e., CD_{ℓ_2} (mm), EMD, $Ot(1)$, and $Ot(2)$. Best results are annotated in **bold**.

Model	3D Point Cloud CCTA Test Dataset			
	$CD_{\ell_2} \downarrow$	$EMD \downarrow$	$Ot(1) \uparrow$	$Ot(2) \uparrow$
PCN	5.04±1.45	3.58±0.94	0.35±0.10	0.55±0.12
SnowflakeNet	3.85±1.16	2.80±0.71	0.43±0.11	0.67±0.12
SPT	3.43±1.41	2.41±0.74	0.47±0.12	0.71±0.13
PointCA ($n_{critic} = 3$)	3.24±1.19	2.31±0.61	0.47±0.14	0.73±0.14
PointCA ($n_{critic} = 1$)	**3.18±1.11**	**2.30±0.60**	**0.49±0.12**	**0.73±0.13**

All values represent mean ± standard deviation.

7.4 Results and Discussion

7.4.1 Analysis on Point Cloud CCTA Test Dataset

The quantitative results on the 3D point cloud CCTA test dataset for our PointCA and three baselines in terms of four metrics, i.e., CD_{ℓ_2}, EMD, $Ot(1)$, and $Ot(2)$, are shown in table 7.1. Our PointCA method achieves the best performance in terms of all four metrics, compared to all other three baselines, i.e., PCN, SnowflakeNet, and SPT. Specifically, our proposed PointCA has an improvement of 7.29%, 4.56%, 4.26%, and 2.82%, in terms of CD_{ℓ_2}, EMD, $Ot(1)$, and $Ot(2)$, respectively, compared to the best baseline model SPT.

We visualise the corresponding point-wise prediction errors in terms of CD_{ℓ_2} between the registered ground truth and 3D point cloud reconstruction for PointCA and three baselines, as illustrated in figure 7.7. We can see that the point cloud reconstruction by our PointCA presents the smallest offsets to ground truth overall compared to the other three baseline models. However, we also find that all models, including our PointCA, struggle to accurately reconstruct sinoatrial nodal arteries (marked by the red boxes), the area of which all models displays apparent offsets relative to the ground truth. More qualitative results on the 3D point cloud CCTA test dataset are illustrated in appendix D.

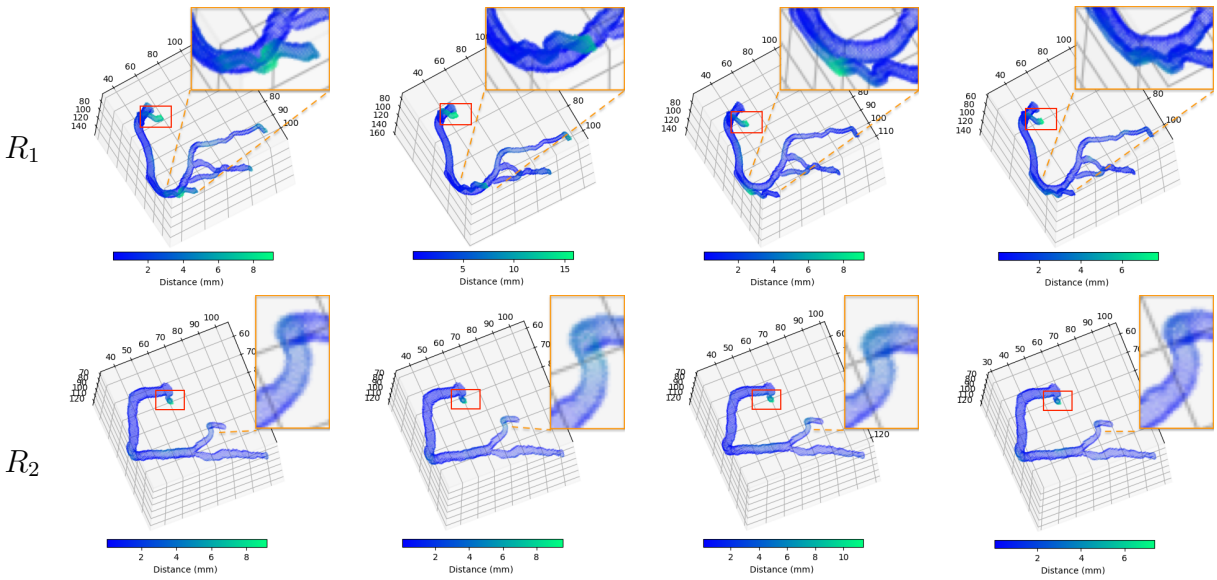


Figure 7.7: Corresponding point-wise prediction errors (mm) in terms of CD_{ℓ_2} between the ground truth and 3D reconstruction results according to their corresponding point spacing. From right to left: visualised prediction errors by our PointCA, SPT, SnowflakeNet, and PCN. From top to bottom: two CCTA test data $R_{1,2}$. The colour bar under each subfigure illustrates the error range for that subfigure.

Ablation Study

We perform three ablation studies on our proposed PointCA method. The first one is to set the number of critic iterations per generator iteration to 3, i.e., $n_{critic} = 3$. The second one considers the model without a conditional adversarial learning module, which is the individual model SPT. The third one replaces the Point Transformer encoder from SPT with the original one, which is the individual model SnowflakeNet. The quantitative results for these three ablation studies on CCTA test dataset are presented in table 7.1. With the Point Transformer encoder added to SnowflakeNet, SPT outperforms SnowflakeNet in all four metrics, which demonstrates the effectiveness of the dynamic point cloud input design to support the different complexities of vessel structures. PointCA ($n_{critic} = 3$) outperforms SPT in three metrics except for $Ot(1)$, where PointCA ($n_{critic} = 3$) has the same mean value but a slightly higher standard deviation. This shows the usefulness of our proposed conditional adversarial learning module. The better performance of PointCA ($n_{critic} = 1$) than PointCA ($n_{critic} = 3$) proves the n_{critic} choice.

Statistical Analysis

We perform the ASO test [205, 206] as implemented by [207] to compare score distributions from different models, which is designed specifically for deep learning models. ASO returns a confidence score ϵ_{min}, which indicates (an upper bound to) the amount of violation of stochastic order. In our work, we set the rejection threshold $\tau = 0.2$ and we choose a significance level $\alpha = 0.05$. We calculate the confidence scores ϵ_{min} of the ASO test for all four models, including both baselines and ablation studies, compared to our proposed PointCA ($n_{critic} = 1$) on CCTA test dataset, in terms of all four metrics, i.e., CD_{ℓ_2}, *EMD*, $Ot(1)$, and $Ot(2)$. The quantitative results of the ASO test are demonstrated in table 7.2.

Table 7.2: Confidence scores ϵ_{min} of ASO test for all four models including both baselines and ablation studies, compared to our proposed PointCA ($n_{critic} = 1$) on CCTA test dataset, in terms of all four metrics, i.e., CD_{ℓ_2}, *EMD*, $Ot(1)$, and $Ot(2)$. Values are in **bold** if $\epsilon_{\text{min}} < \tau = 0.2$.

Model	**3D Point Cloud CCTA Test Dataset**			
	CD_{ℓ_2}	*EMD*	$Ot(1)$	$Ot(2)$
PCN	**<1e-4**	**<1e-4**	**1.42e-3**	**2.12e-3**
SnowflakeNet	**<1e-4**	**<1e-4**	**0.09**	**0.13**
SPT	**<1e-4**	**<1e-4**	0.46	0.36
PointCA ($n_{critic} = 3$)	**<1e-4**	**<1e-4**	0.61	0.68

As demonstrated in table 7.2 with the rejection threshold $\tau = 0.2$, we are confident that our PointCA ($n_{critic} = 1$) is tested to be stochastically dominant over both PCN and SnowflakeNet models in terms of all four metrics. Regarding models SPT and PointCA ($n_{critic} = 3$), we are confident that our PointCA ($n_{critic} = 1$) is tested to be stochastically dominant over them in terms of CD_{ℓ_2} and *EMD*. The confidence scores ϵ_{min} of the ASO test for model SPT compared to PointCA ($n_{critic} = 1$) in terms of $Ot(1)$ and $Ot(2)$ are less than 0.5, which means PointCA ($n_{critic} = 1$) is better than model SPT in most cases. Overall, the quantitative results statistically indicate our PointCA's superior ability in coronary artery tree reconstruction compared to both baseline models and ablation studies.

Table 7.3: Quantitative results for our PointCA method and three baselines, i.e., PCN, SnowflakeNet, and SPT, evaluated on two projection planes and one additional unseen projection plane of 2D clinical ICA data (unseen domain) in terms of three metrics, i.e., $Ot(1)$, $Ot(2)$, and CD_{ℓ_2} (mm). Best results are annotated in **bold**.

Model	2D Real Clinical ICA Dataset (Unseen Domain)								
	1^{st} Projection			2^{nd} Projection			Additional Unseen Projection		
	$Ot(1)\uparrow$	$Ot(2)\uparrow$	$CD_{\ell_2}\downarrow$	$Ot(1)\uparrow$	$Ot(2)\uparrow$	$CD_{\ell_2}\downarrow$	$Ot(1)\uparrow$	$Ot(2)\uparrow$	$CD_{\ell_2}\downarrow$
PCN	0.54±0.05	0.65±0.05	4.57±1.15	0.48±0.06	0.58±0.06	**5.55±1.07**	0.51±0.02	0.61±0.02	5.86±0.17
SnowflakeNet	0.57±0.08	0.69±0.08	4.91±1.81	0.48±0.08	0.59±0.09	6.37±1.72	0.49±0.02	0.61±0.02	6.40±0.33
SPT	0.60±0.08	**0.71±0.07**	4.39±1.70	**0.51±0.10**	**0.62±0.11**	5.74±2.47	0.50±0.04	0.61±0.03	6.04±0.02
PointCA	**0.60±0.07**	**0.71±0.07**	**4.20±1.75**	0.50±0.06	0.61±0.07	5.56±1.10	**0.55±0.10**	**0.66±0.12**	**4.87±1.85**

All values represent mean ± standard deviation.

7.4.2 Analysis on Clinical ICA Dataset

The quantitative results for our PointCA method and three baselines, evaluated on two projection planes and one additional unseen projection plane of 2D clinical ICA data (unseen domain) in terms of three metrics, i.e., $Ot(1)$, $Ot(2)$, and CD_{ℓ_2}, are presented in table 7.3. We can observe that our proposed PointCA attains the best performance in all three metrics on both the first and additional projection planes compared to three baselines, i.e., PCN, SnowflakeNet, and SPT. For the second projection plane, our method achieves the second best performance, only 1.96%, 1.61%, and 0.18% behind the best baseline models in terms of $Ot(1)$, $Ot(2)$, and CD_{ℓ_2}, separately. The results for the additional unseen projection plane are the most significant, since the model is trained and inferred only based on two projections. On the additional unseen projection plane, our proposed PointCA achieves an improvement of 7.84%, 8.20%, and 16.89%, in terms of $Ot(1)$, $Ot(2)$, and CD_{ℓ_2}, respectively, compared to the best baseline models. This indicates our PointCA method has a better generalisation ability to the unseen domain.

7.4.3 Comparison with Voxel-based Methods on ICA Data

To the best of our knowledge, DeepCA and IterCA are the only two deep learning-based methods to solve 3D coronary artery tree reconstruction from two clinical non-simultaneous ICA projections, but they are based on voxel representations. We present the quantitative comparison between our PointCA and the two voxel-based deep learning methods, i.e., DeepCA and IterCA, on the unseen real clinical ICA data, as illustrated in table 7.4. We

only compare them based on the second and additional projection planes since there is no motion on the first projection plane for voxel-based methods. We can observe that, apart from the results of metric $Ot(1)$ on the second projection, our PointCA has worse results than the voxel-based methods. This may be due to the fact that point cloud reconstruction is easily deformed because point clouds are not confined to fixed positions like voxel grids, which would indicate an unfair direct quantitative comparison between point cloud-based and voxel-based methods.

Table 7.4: Quantitative results on two projection planes for our PointCA method and two voxel-based baselines, i.e., DeepCA and IterCA, on clinical ICA dataset (unseen domain), in terms of two metrics, i.e., $Ot(1)$ and CD_{ℓ_2} (mm).

Model	2D Real Clinical ICA Dataset (Unseen Domain)			
	2^{nd} Projection		Additional Unseen Projection	
	$Ot(1)\uparrow$	$CD_{\ell_2}\downarrow$	$Ot(1)\uparrow$	$CD_{\ell_2}\downarrow$
DeepCA	0.46±0.07	4.51±1.29	0.59±0.04	2.81±0.06
IterCA	0.53±0.09	3.88±1.36	0.63±0.07	2.58±0.88
PointCA	0.50±0.06	5.56±1.10	0.55±0.10	4.87±1.85

All values represent mean ± standard deviation.

For qualitative comparison with voxel-based methods, we scale up the reconstruction results of DeepCA and IterCA to the same high-resolution space as the point cloud reconstruction results by our PointCA. The 3D visualisation on two clinical ICA samples is illustrated in figure 7.8. We can see that our PointCA can reconstruct the coronary artery tree with dense information due to the high efficiency of point cloud-based representation, while voxel-based methods DeepCA and IterCA produce sparse coronary artery tree structures when considering a high-resolution space. We also find that the overall structures of the reconstructed coronary artery trees by our PointCA and DeepCA are more similar than those of IterCA. This is possibly due to the fact that IterCA leverages iterative registration to perform reconstruction, which results in reduced deformation in 3D reconstruction compared to one-time reconstruction methods, PointCA and DeepCA. Overall, the successful 3D reconstruction results by PointCA from real clinical unseen ICA data demonstrate that through simulating projections from CCTA point clouds, we are able to implicitly

compensate for the non-rigid motion between non-simultaneous projections to enable 3D reconstruction. More qualitative results of point cloud reconstruction on clinical ICA data by our PointCA are presented in appendix D.

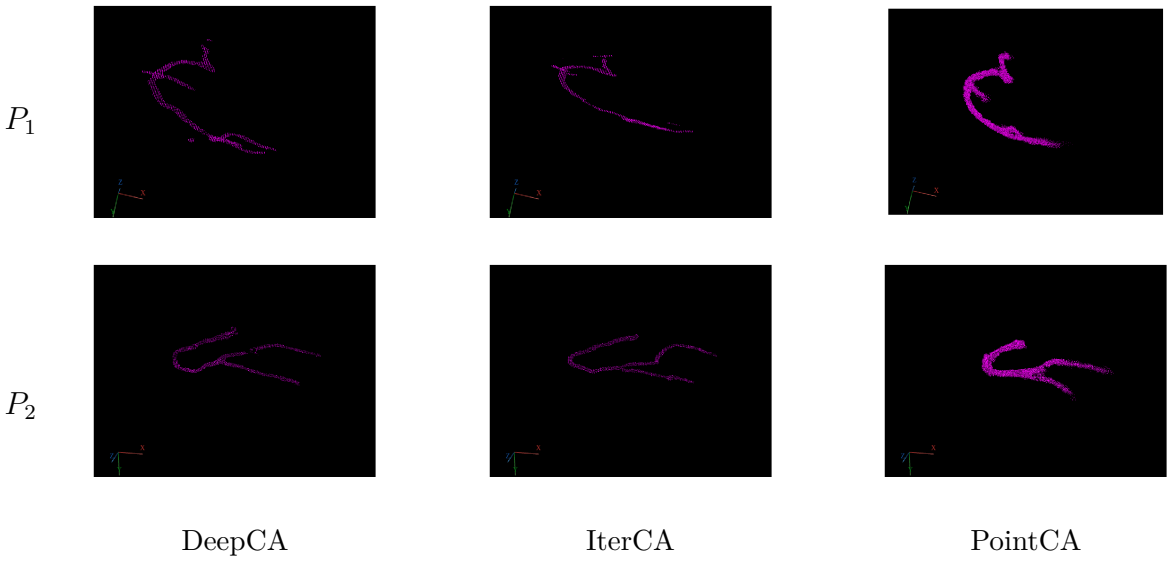


Figure 7.8: Reconstruction results by our PointCA, IterCA and DeepCA methods (right to left) on two real clinical data $P_{1,2}$. The reconstruction results of IterCA and DeepCA are scaled to the same high-resolution space as the point cloud reconstruction results by our PointCA before visualisation.

We do not have 3D ground truth for real clinical ICA data, so we additionally display the reprojections of 3D reconstruction by our PointCA, IterCA, and DeepCA, together with the corresponding original ICA data, as illustrated in figure 7.9. Similarly, the reconstruction results of IterCA and DeepCA are scaled to the same high-resolution space as the point cloud reconstruction results by our PointCA before performing reprojection. We can see the same results as in the 3D qualitative analysis: the vessel structures in the reprojections from DeepCA and IterCA are more sparse compared with our PointCA. However, compared with the original ICA data, we observe that DeepCA and IterCA can reconstruct more nuanced details than our PointCA. This may be because our PointCA has to generate a large amount of points, so it is less easy to capture nuanced distinctions considering the complex tortuous shape of coronary vessels.

We further show the distributions of ground truth surface sizes between point clouds used in our PointCA and volume grids used in voxel-based methods, as illustrated in

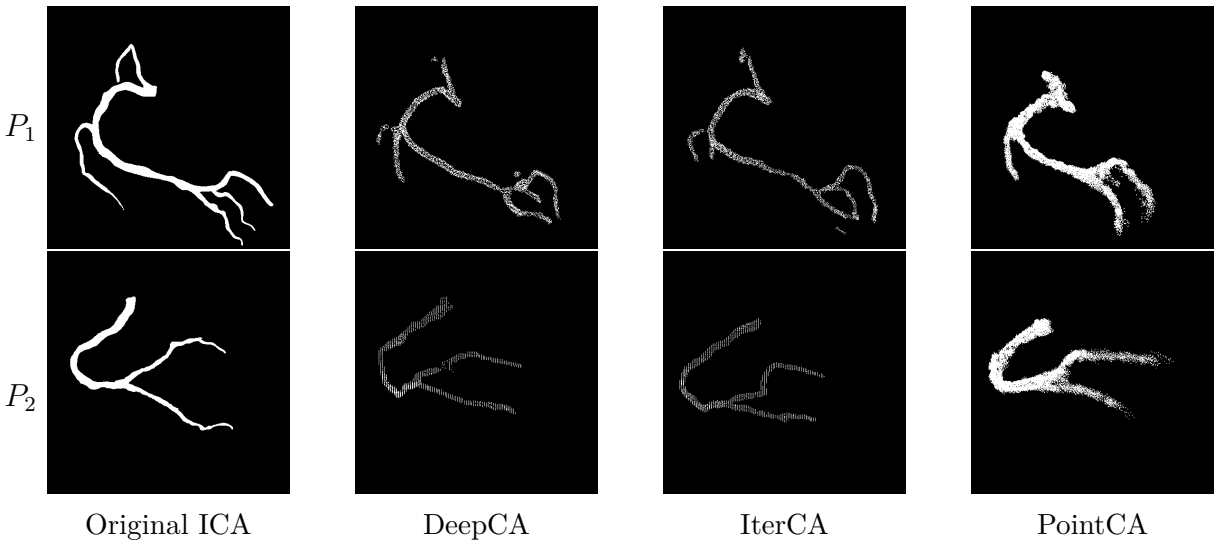
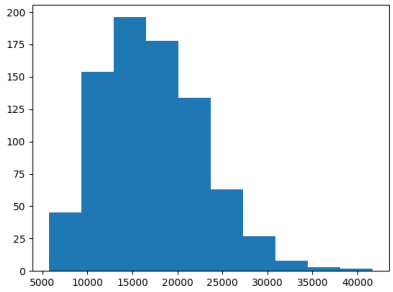
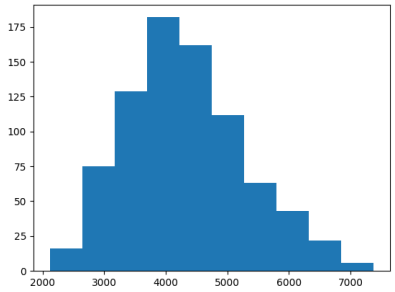


Figure 7.9: Reprojection results on the second and additional projection planes (bottom to top) by our PointCA, IterCA, and DeepCA (right to left) evaluated on two real clinical data $P_{1,2}$. The first column presents the corresponding original ICA data. The reconstruction results of IterCA and DeepCA are scaled to the same high-resolution space as the point cloud reconstruction results by our PointCA before performing reprojection.

figure 7.10, where the surface of ground truth point clouds used in our PointCA maintains CCTA's original high resolution. We can see that the average ground truth surface size of the point cloud is around 4 times that of the volume grids. This means our point cloud-based method can learn more information owing to its efficient data representation. Based on this information, we set our PointCA to produce a fine reconstruction point cloud of size $18,432$ to approximately match the average point cloud ground truth surface size, as described in section 7.2.1. Compared to the surface size of volume grids used in voxel-based methods, the reconstructed point cloud with size $18,432$ from our PointCA is very dense, which has been qualitatively illustrated in figures 7.8 and 7.9. Moreover, due to the efficient data representation of point clouds, our PointCA is trained based on a 32 GB GPU with a batch size of 15, while DeepCA and IterCA backbone were trained based on a 48 GB GPU with a batch size of 3. However, we should still notice the limitation of nuanced details recovery in our PointCA as qualitatively discussed before, so further improvement is required to produce accurate dense coronary artery tree surface reconstruction with precise structural details.



Point Cloud



Volume Grid

Figure 7.10: Distributions of ground truth surface size between point clouds used in our PointCA (left) and volume grids used in voxel-based methods (right). Vertical axis stands for the number of ground truth data and horizontal axis represents the surface size.

7.5 Conclusion

In this chapter, we propose a novel point cloud-based method, named PointCA, to efficiently reconstruct dense coronary artery trees from two 2D non-simultaneous ICA projections. We introduce a novel network Snowflake Point Transformer and a novel Wasserstein conditional point adversarial learning module with gradient penalty designated for distinguishing point cloud structures. The dynamic input design of our model allows its application in varied datasets. Through simulating projections from the CCTA data, the non-rigid motion between two non-simultaneous projections is implicitly compensated to facilitate 3D reconstruction. We evaluate our proposed PointCA on a public CCTA dataset and a real ICA dataset (unseen domain), in comparison to three other models. The results demonstrate the promising performance of our PointCA method in dense point cloud-based coronary artery tree reconstruction while largely reducing computational inefficiency.

Chapter 8

Conclusion and Future Works

8.1 Conclusion

In this Thesis, we have designed, implemented and validated automated 3D coronary artery tree reconstruction algorithms based on deep learning techniques, using only two non-simultaneous ICA projections, thus mitigating potential chemotoxic adverse reactions and radiation harm, as well as reducing the risks of subjective interpretation of the 3D vasculature from 2D views.

Our research journey started with the investigation of the feasibility of reconstructing a 3D coronary artery tree from two projections without any motion between projection planes. We leveraged the idea of implicit neural representation from NeRFs and proposed our self-supervised deep learning method, NeCA, to achieve the reconstruction goal. Our proposed NeCA utilises a learnable multiresolution hash encoder and a differentiable projector layer to enable effective feature encoding and self-supervised learning from 2D input projections. It neither requires 3D ground truth for supervision nor large training datasets and optimises the reconstruction results in an iterative self-supervised fashion with only the projection data of one patient as input. Compared to the supervised model, our proposed model demonstrates promising performance in both vessel topology preservation, recovery of missing features, and branch-connectivity maintenance, which proves the practicality of 3D coronary artery tree reconstruction from only two projections using deep learning.

Clinical ICA images are usually acquired non-simultaneously, so there is motion between projections, such as the non-rigid cardiac and respiratory motions. Due to this, we next explored the reconstruction scenarios under two non-simultaneous projections. This

challenge is further complicated by the problems of both the limited number of real paired ICA data with projection geometry information and the unavailable 3D ground truth for real ICA data. We found the solution by simulating projections from data of a public CCTA dataset, with a rigid transformation added for the second projection plane. We proposed DeepCA, a deep learning model based on the WCGAN with gradient penalty, latent convolutional transformer layers, and a dynamic snake convolutional critic. Via mapping the input with two non-aligned simulated projections to 3D coronary artery tree data, most motion artifacts are corrected by our model. With the critic used, any residual uncorrected deformations are adjusted, while ensuring the connectedness of the coronary tree structures in the reconstructions and increasing the model's elastic generalisation capacity. So when generalising to two real clinical non-simultaneous ICA projections, the non-rigid motion between projections is compensated implicitly to provide promising 3D coronary artery tree reconstruction. We also provided an application-specific evaluation method to tackle the deformation in 3D reconstructions, the unavailability of 3D ground truth for real ICA scans, and the motion between projection planes. To the best of our knowledge, this is the first study that leverages deep learning in 3D coronary artery tree reconstruction from two real clinical non-simultaneous ICA projections, which thus provides a baseline for future work in this area.

Based on the idea of simulating projections, we further checked four different misalignment levels of rigid transformations simulated on the public CCTA dataset for approximating typical misalignment amounts found in clinical acquisitions for training. We selected one trained model, evaluated as the most accurate across all misalignment levels, for better generalisation to various unseen real-world scenarios. We then came up with an iterative registration strategy and proposed IterCA, whose backbone is based on DeepCA, to iteratively explicitly compensate for motion on the non-simultaneous projection plane to refine 3D coronary artery tree reconstruction. We performed an extensive evaluation by validating our proposed IterCA on the CCTA test dataset and three unseen datasets, including a different CCTA dataset, a synthetic dataset, and a real ICA dataset. The results

show that through iterative registration, IterCA can gradually eliminate the negative effect of large motion and align the two non-simultaneous projection planes, enabling stable reconstruction under different conditions and showing better generalisation capability.

The proposed DeepCA and IterCA solve the problem of reconstruction from non-simultaneous projections, but they rely on voxel-based representations, which are inefficient for the reconstruction of sparse, complex curvilinear structures such as coronary vessel trees, and make it hard to perform high-resolution reconstruction due to the limitations of computational resources. Therefore, as the final contribution, we developed and validated a point cloud-based approach, PointCA, to efficiently handle vessel topology, for dense 3D coronary artery tree surface reconstruction from two 2D non-simultaneous ICA projections. We introduced a network, Snowflake Point Transformer, and a point adversarial learning module designed for distinguishing point cloud structures. Our method supports a dynamic size of different point clouds in one batch, allowing various vasculature reconstructions. The results demonstrate the promising performance of our PointCA method in dense point cloud coronary artery tree reconstruction while largely reducing computational inefficiency.

In conclusion, this Thesis presents progressive steps towards 3D coronary artery reconstruction, from simulated projections without motion to real clinical ICA data with potential non-rigid cardiac motion, from unstable to stable reconstruction on the unseen clinical ICA data, and from inefficient to efficient high-resolution reconstruction. The thesis not only introduces novel methodologies to tackle the reconstruction problem, but also provides details of the necessary data simulation process and both crucial application-specific evaluation procedures and metrics. It can advance our understanding of how to handle an ill-posed problem based on limited unseen medical data. A fully automated system could build the 3D coronary artery tree during cardiac interventions, to reduce the risks of subjective interpretation of 3D coronary vasculature from 2D views, assist pre-operative planning, provide intra-operative direction and non-invasively measure physiological indices. This could dramatically ease the complexity of real-time diagnosis and interventional surgeries.

8.2 Future Works

8.2.1 Fully End-to-end Reconstruction from ICA Data

None of our proposed methods directly use the original clinical ICA data, since we have manually segmented the coronary artery structures from clinical ICA data for our studies. There are several methods for automated coronary vessels segmentation based on deep learning [238, 239], which are essential to achieve fully end-to-end 3D coronary artery tree reconstruction during cardiac interventions. However, these methods' stability and performance gap compared to manual segmentation should be carefully considered.

8.2.2 Foundation Models or Large Language Models (LLMs) as Strong Generative Backbones

Accurate coronary artery tree reconstruction is critical for clinical use. If incorrect, it could have serious consequences, such as selecting an inappropriate stent size, detecting stenosis by mistake, or finding the wrong stenosis location. Our proposed methods can now produce 3D coronary artery tree reconstruction with either satisfactory vessel structures or high resolution, but not both. This is probably due to the limited information from two projections. Recently, foundation models and large language model (LLM), in particular the large vision-language model (VLM), have demonstrated strong generative abilities [86, 87, 104], which have been used as a strong pretrained backbone for many downstream tasks. For example, these large-scale models are able to generate novel views and infer various 3D features, given specific prompts which are generated based on available limited projections. This is helpful to decrease the difficulty of reconstruction from limited two projections and improve the reconstruction quality. However, constructing correct prompts is crucial for the success of later reconstruction. Moreover, the foundation models and VLMs may not have 'seen' plenty of medical images, especially angiographic projections, which may limit their generative capability in our task. We are not likely to use large medical datasets for training large-scale models, since this requires expensive computational

resources and large medical datasets are hard to acquire. We can adopt some adaptation strategies for large models to fine-tune them to solve our new, specific tasks, using a small amount of medical data in our area.

8.2.3 Next-step Clinical Validation

Clinical validation is essential to meet the stringent clinical standard. Our current evaluation on clinical ICA data is compromised due to the absence of 3D motion-free ground truth. After developing an automated model for the task, we must perform a robust clinical evaluation before clinical use. We can use another 3D modality, OCT, to facilitate the clinical validation. OCT is an invasive medical imaging modality, providing high-resolution endoluminal visualisation of the lesion anatomy [260]. We can collect OCT data for the same patient in our reconstruction task. The cross-sections from our reconstruction results should match the OCT derived luminal cross-sections of the same patient. The OCT evaluation allows accurate comparison of vessel lumen cross-sections to see if there is any significant difference in reconstruction performance, which is effective for the evaluation of our coronary artery tree surface reconstruction. Apart from the additional OCT data for validation, there is a need for various ICA data from multi centres to prove the method's effectiveness, safety, and generalisability across different patient populations and imaging equipment.

Our current contributions in the thesis all focus on anatomical reconstruction, while one of our clinical goals is functional assessment which remains to be solved. A key gap is seamlessly integrating the 3D geometric reconstruction with computational fluid dynamics simulation to provide a comprehensive diagnostic tool, such as measuring FFR. We should demonstrate the clinical utility of our automated tool, such as whether an automated 3D reconstruction tool reduces restenosis rates by selecting the appropriate stent size and reduces the death rates by timely performing angioplasty on the severe stenosis whose severity is determined by FFR measurement.

Appendix A

Neural Implicit Representation

More Qualitative Results - 3D Coronary Artery Tree Reconstruction from Two 2D Projections via Neural Implicit Representation

We present aditional qualitative results of 3D coronary artery tree reconstruction based on our NeCA model, NeCA (90°), and the 3D U-Net model on both the RCA and LAD test datasets.

RCA Dataset

3D Reconstruction Results

Figure A.1 illustrates additional five RCA examples of 3D coronary artery tree reconstruction using our NeCA model, NeCA (90°), and 3D U-Net model, along with the corresponding ground truth for each case. The results show that all three models can successfully perform satisfactory 3D RCA reconstruction.

Comparison Between 3D Reconstruction and Ground Truth

We additionally compare the five 3D RCA reconstruction results using the NeCA, NeCA (90°), and 3D U-Net model with the corresponding ground truth in the same 3D space, as illustrated in figure A.2. These figures show that our NeCA model demonstrates better reconstruction overlap than the 3D U-Net model.

Visualisation of Chamfer ℓ_2 Distance (CD_{ℓ_2})

We visualise the corresponding voxel-wise prediction errors (mm) in terms of CD_{ℓ_2} between the ground truth and 3D reconstruction results with a binarisation threshold of 0.5 based on ten RCA data according to their corresponding voxel spacing, with respect to our NeCA model, our NeCA model using orthogonal projections, and 3D U-Net model, as illustrated in figures A.3 and A.4.

LAD Dataset

3D Reconstruction Results

We show in figure A.5 additional five 3D LAD reconstruction results using our NeCA model, NeCA (90°), and the 3D U-Net model, with the corresponding ground truth. From the results, we can observe that our NeCA model successfully reconstructs the LAD vasculature in all the cases. On the other hand, the 3D U-Net model fails to reconstruct some branches in L_7 and loses vessel connectivity, as presented in red boxes.

Comparison Between 3D Reconstruction and Ground Truth

We also compare in figure A.6 the additional five 3D LAD reconstruction results using NeCA, NeCA (90°), and the 3D U-Net models with the corresponding ground truth in the same 3D space. The results show similar performance to the RCA dataset; our NeCA model demonstrates better reconstruction overlap than the 3D U-Net model.

Visualisation of Chamfer ℓ_2 Distance (CD_{ℓ_2})

The corresponding voxel-wise prediction errors (mm) regarding CD_{ℓ_2} between the ground truth and 3D reconstruction results with a binarisation threshold of 0.5 based on ten LAD data according to their corresponding voxel spacing are illustrated in figures A.7 and A.8, in terms of our NeCA model, our NeCA model using orthogonal projections, and 3D U-Net model.

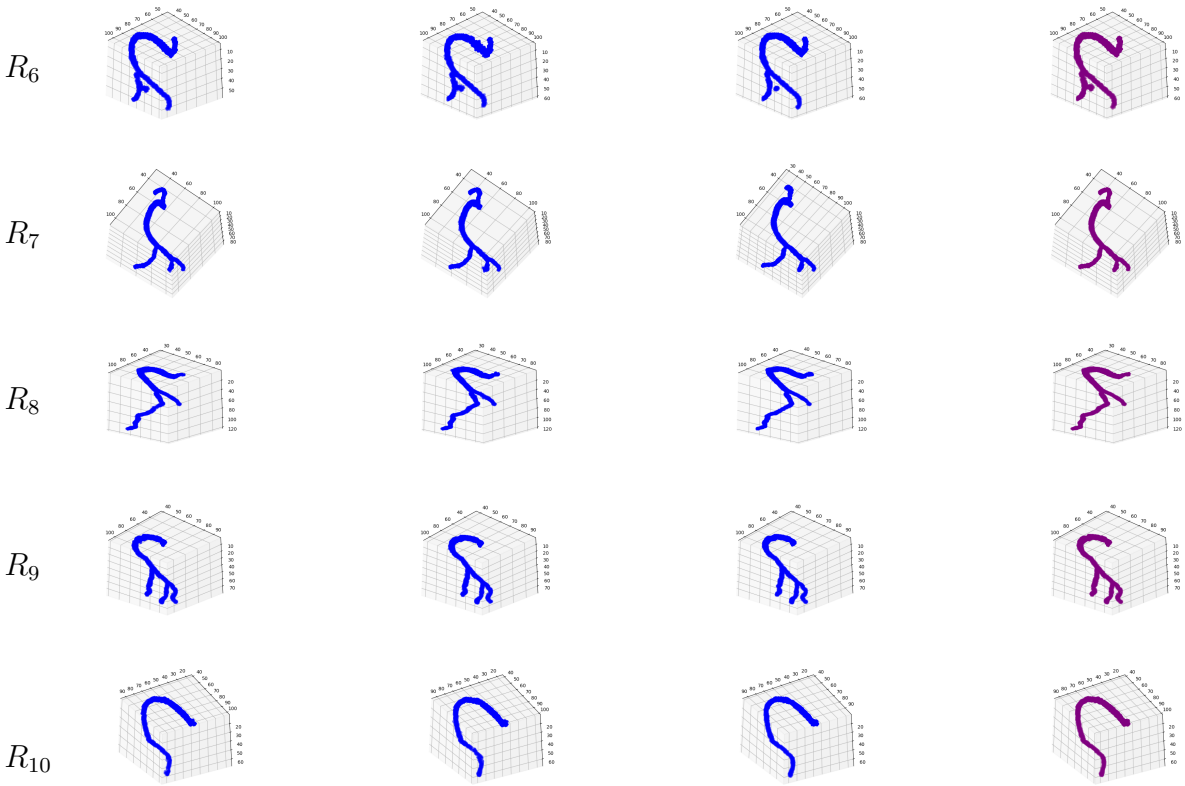


Figure A.1: Five qualitative results of 3D RCA reconstruction with a binarisation threshold of 0.5. From top to bottom: five RCA data points $R_{6,7,8,9,10}$. From left to right: the reconstruction results from our NeCA model, our NeCA model using two orthogonal projections (90°), and 3D U-Net model, along with the corresponding ground truth.

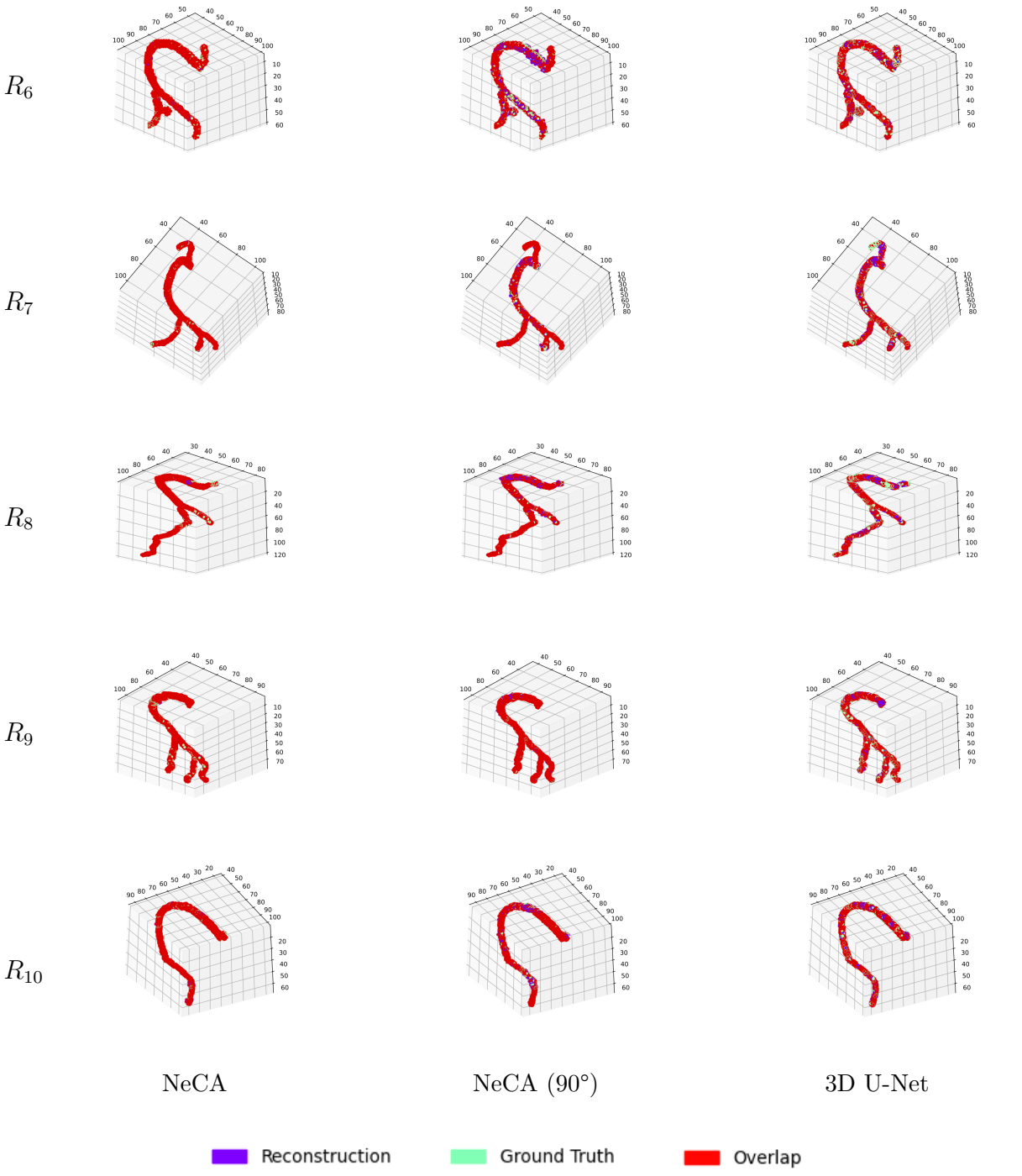


Figure A.2: Five 3D RCA reconstruction results by a binarisation threshold of 0.5 compared with the corresponding ground truth in the same 3D space. From top to bottom: five RCA data $R_{6,7,8,9,10}$. From left to right: the comparison results from our NeCA model, our NeCA model using two orthogonal projections (90°), and 3D U-Net model. Colour purple represents the reconstruction results, colour green is the ground truth, and colour red means the overlap between them.

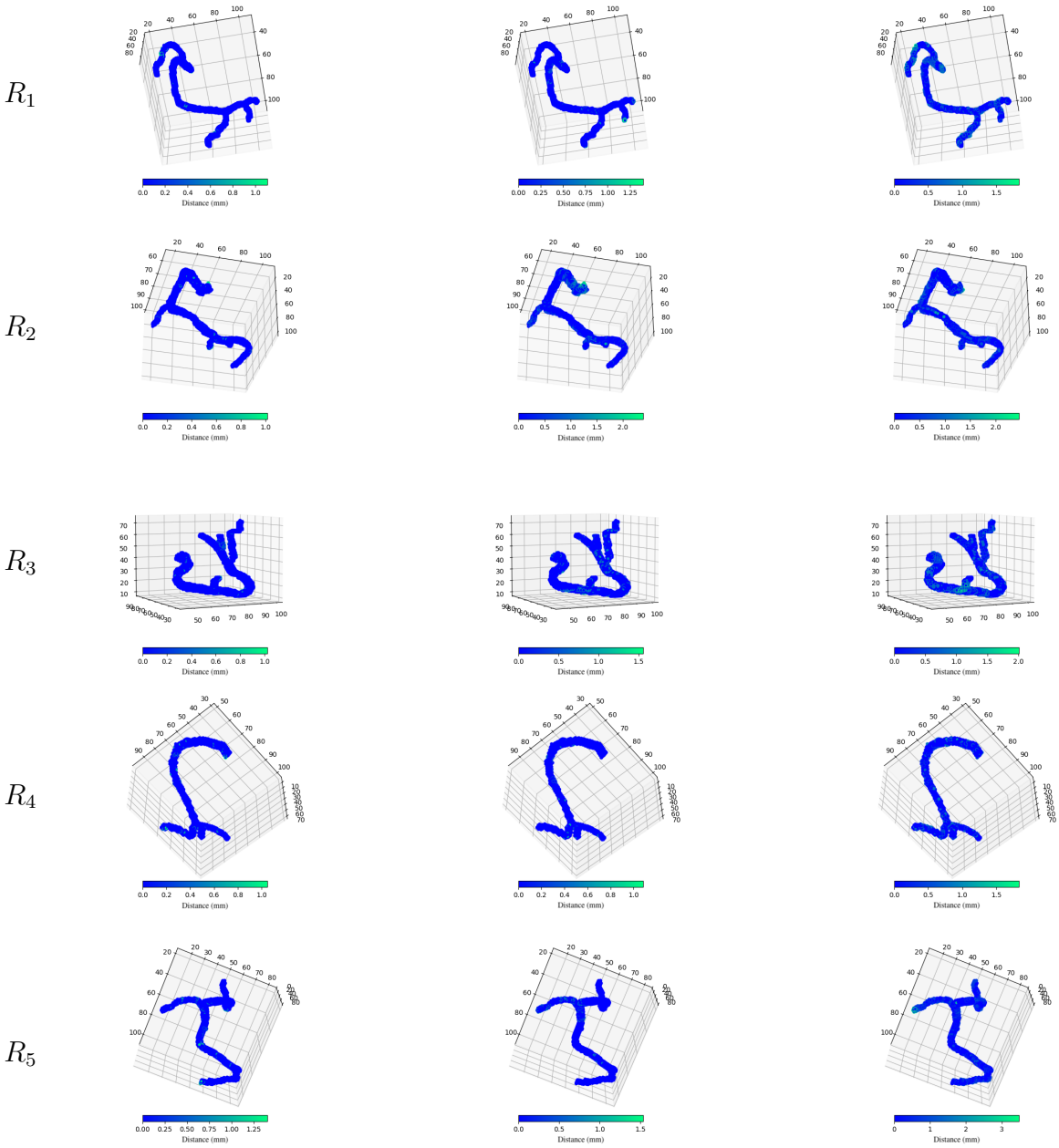


Figure A.3: Five RCA qualitative examples of the corresponding voxel-wise prediction errors (mm) in terms of CD_{ℓ_2} between the ground truth and 3D reconstruction results with a binarisation threshold of 0.5 according to their corresponding voxel spacing. From top to bottom: five RCA data $R_{1,2,3,4,5}$. From left to right: the prediction errors from our NeCA model, our NeCA model using two orthogonal projections (90°), and 3D U-Net model. The colour bar under each subfigure illustrates the error range for that subfigure.

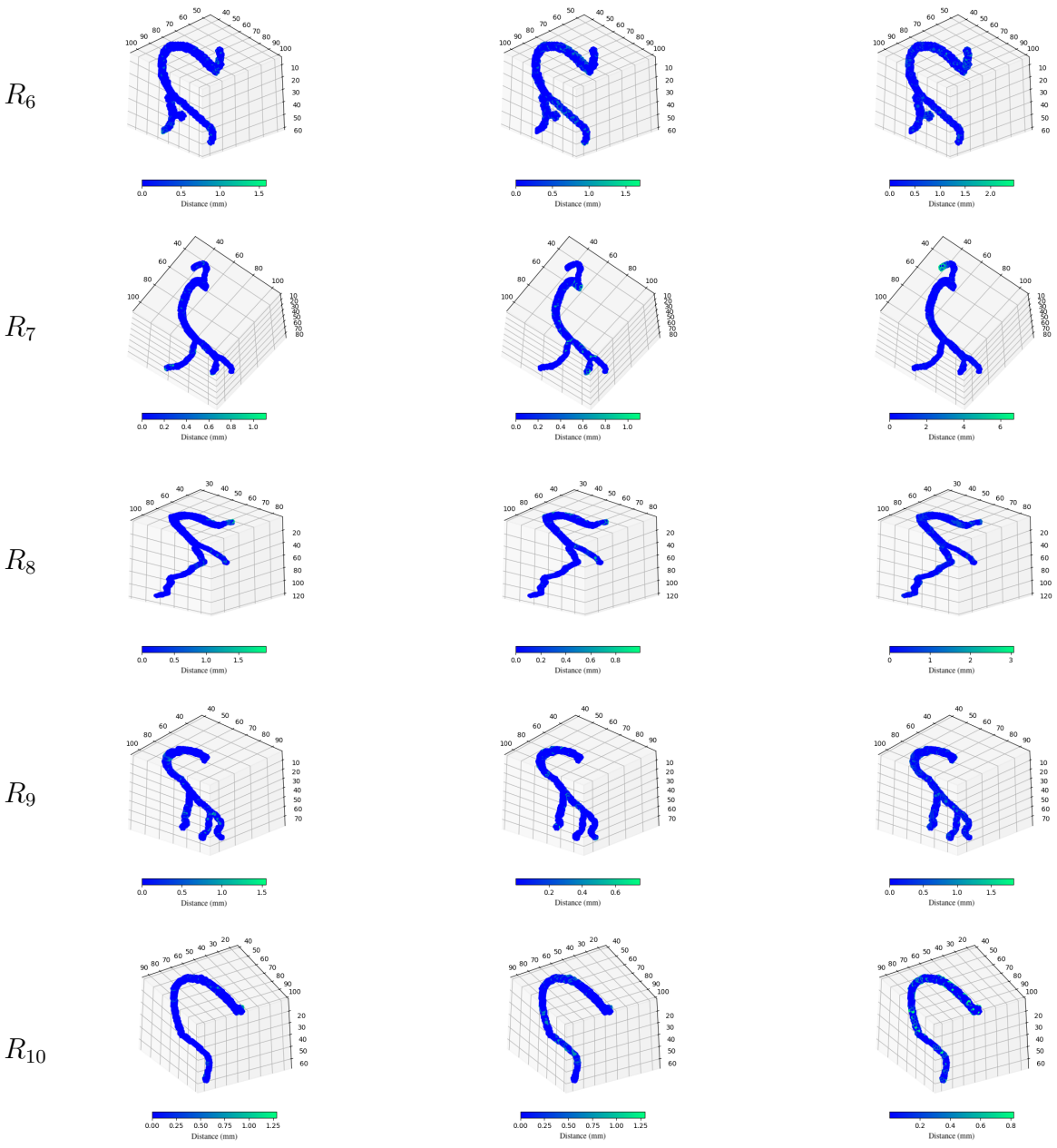


Figure A.4: Five RCA qualitative examples of the corresponding voxel-wise prediction errors (mm) in terms of CD_{ℓ_2} between the ground truth and 3D reconstruction results with a binarisation threshold of 0.5 according to their corresponding voxel spacing. From top to bottom: five RCA data $R_{6,7,8,9,10}$. From left to right: the prediction errors from our model, our model using two orthogonal projections (90°), and 3D U-Net model. The colour bar under each subfigure illustrates the error range for that subfigure.

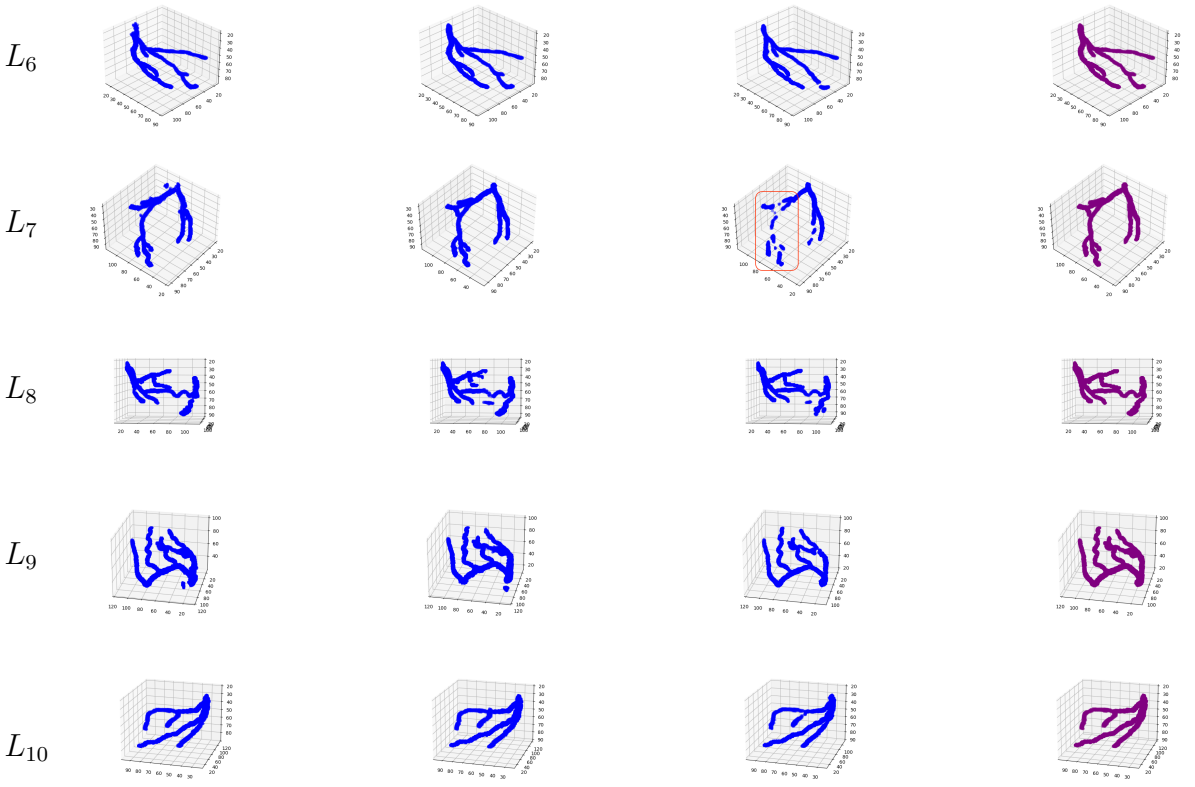


Figure A.5: Five qualitative 3D LAD reconstruction results with a binarisation threshold of 0.5. From top to bottom: five LAD data $L_{6,7,8,9,10}$. From left to right: the reconstruction results from our NeCA model, our NeCA model using two orthogonal projections (90°), and 3D U-Net model, along with the corresponding ground truth.

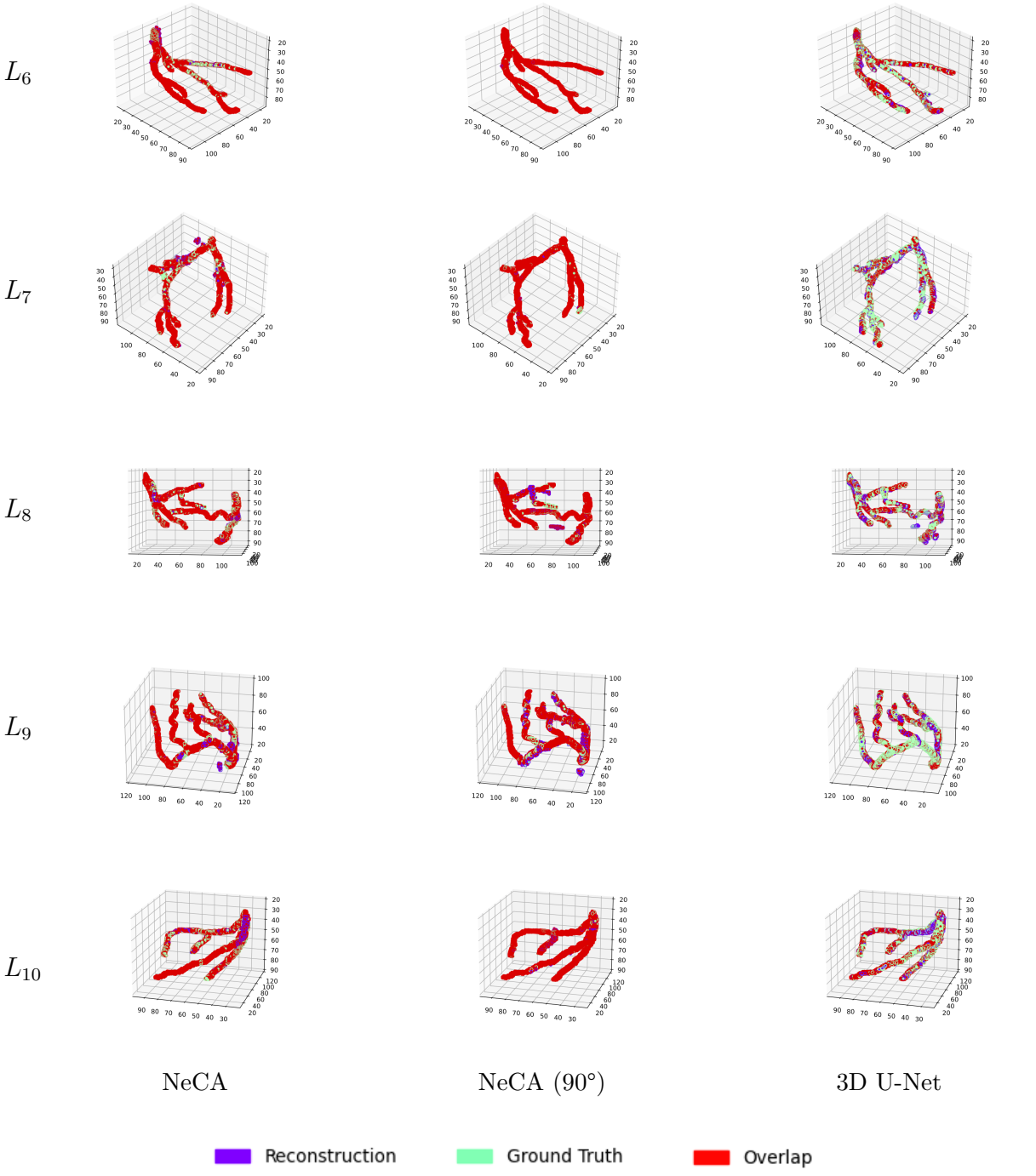


Figure A.6: Five 3D LAD reconstruction results by a binarisation threshold of 0.5 compared with the corresponding ground truth in the same 3D space. From top to bottom: five LAD data $L_{6,7,8,9,10}$. From left to right: the comparison results from our NeCA model, our NeCA model using two orthogonal projections (90°), and 3D U-Net model. Colour purple represents the reconstruction results, colour green is the ground truth, and colour red means the overlap between them.

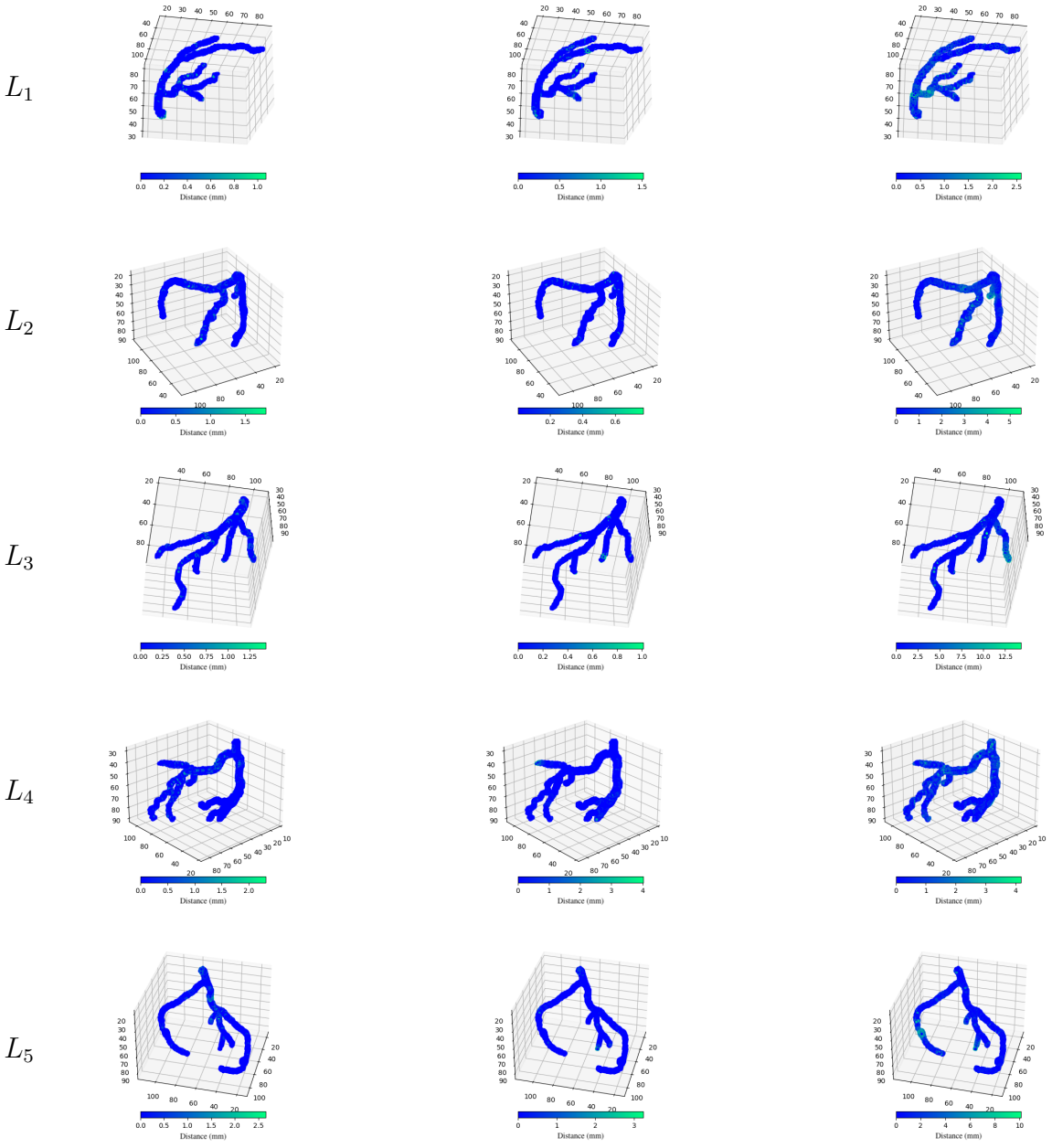


Figure A.7: Five LAD qualitative examples of the corresponding voxel-wise prediction errors (mm) in terms of CD_{ℓ_2} between the ground truth and 3D reconstruction results with a binarisation threshold of 0.5 according to their corresponding voxel spacing. From top to bottom: five LAD data $L_{1,2,3,4,5}$. From left to right: the prediction errors from our NeCA model, our NeCA model using two orthogonal projections (90°), and 3D U-Net model. The colour bar under each subfigure illustrates the error range for that subfigure.

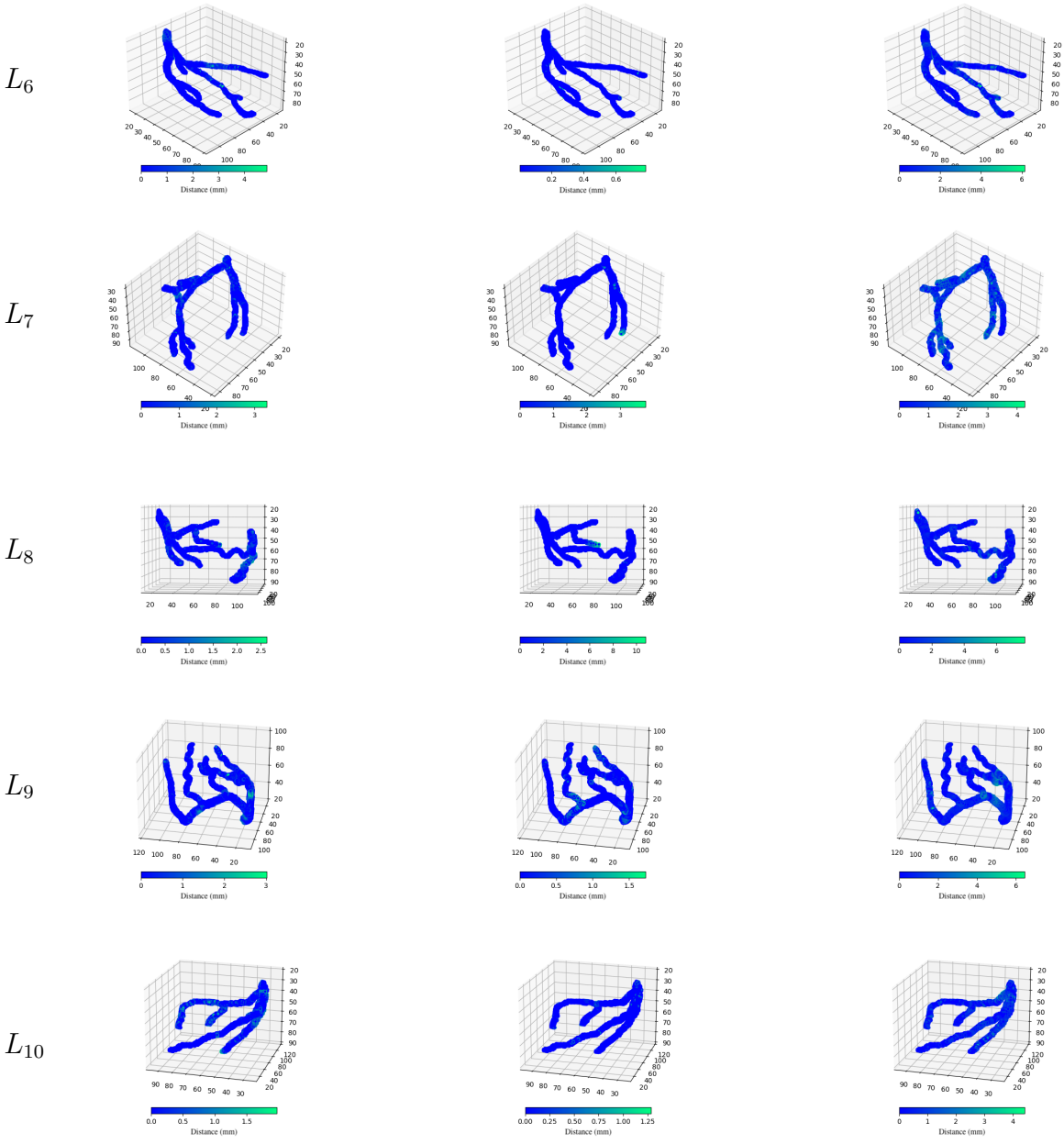


Figure A.8: Five LAD qualitative examples of the corresponding voxel-wise prediction errors (mm) in terms of CD_{ℓ_2} between the ground truth and 3D reconstruction results with a binarisation threshold of 0.5 according to their corresponding voxel spacing. From top to bottom: five LAD data $L_{6,7,8,9,10}$. From left to right: the prediction erros from our NeCA model, our NeCA model using two orthogonal projections (90°), and 3D U-Net model. The colour bar under each subfigure illustrates the error range for that subfigure.

Appendix B

Reconstruction on Clinical Data

More Qualitative Results - 3D Coronary Artery Tree Reconstruction from Two 2D Non-simultaneous X-ray Angiographic Projections

We show additional qualitative results for our proposed DeepCA model, 4 baseline models, and 3 ablation models on the CCTA dataset and all the ICA data.

Qualitative Results on 3D CCTA Test Dataset

We present 5 CCTA test data samples for qualitative analysis. Before evaluation, we rigidly register the ground truth (original CCTA test data) to the 3D reconstruction. Figure B.1 shows the original ground truth and the 3D reconstructions generated by all the models visualised from the front view. Figure B.2 illustrates the corresponding voxel-wise prediction errors in terms of CD_{ℓ_2} between the ground truth and the 3D reconstruction.

Qualitative Results on 2D Clinical ICA Dataset

We present the 3D reconstructions by all the models and the corresponding 2D reprojections when testing on the unseen 2D real clinical ICA dataset of 8 patients. Figure B.3 displays the 3D reconstructions by all the models. Figure B.4 illustrates the comparisons on the first projection plane between the original ICA data and the reprojections. Before evaluation on the second and additional projection planes, we first rigidly register the original ICA data to the reprojections. Figures B.5 and B.6 present the comparisons on the second and additional projection planes between the registered ICA data and the reprojections.

Discussion

We can see from figures B.1 and B.2 that our proposed DeepCA model has successfully reconstructed all the branches and maintained the vessel connectivity, while for the baseline models, there are many missing and/or broken branches visible. Although there is no corresponding 3D ground truth for the real clinical ICA data, we observe the same results in figure B.3. In particular, some 3D reconstruction results on clinical ICA data from the baseline models miss the vascular features almost entirely, such as the reconstructions by model Un2+ and Un3+ on patients 2, 3, and 8.

The qualitative evaluation results on all three projection planes, as illustrated in figures B.4 to B.6, demonstrate the superiority of our proposed DeepCA model's performance on real clinical ICA data as well. These results indicate that our proposed DeepCA model has the best performance in vessel topology preservation and recovery of missing features.

Moreover, in all the qualitative results from the 3 ablation models, we can find that each component of our proposed DeepCA model has contributed to the final reconstruction performance with more vascular features recovered and broken branches connected.

Figure B.1: 3D reconstruction results on 5 CCTA test data from all the models. From left to right: 5 CCTA test data samples. Row 1: 3D ground truth. From row 2 to the end: 3D reconstruction results by our proposed DeepCA model, WGP, +CTLs, +DSCC, Un2+, Un3+, DSCN, and CVTG.

Figure B.2: Corresponding voxel-wise prediction errors in terms of CD_{ℓ_2} between the ground truth and 3D reconstruction, after rigidly registering the ground truth to reconstructions from all the models. From left to right: 5 CCTA test data samples. From top to bottom: prediction errors by our proposed DeepCA model, WGP, +CTLs, +DSCC, Un2+, Un3+, DSCN, and CVTG.

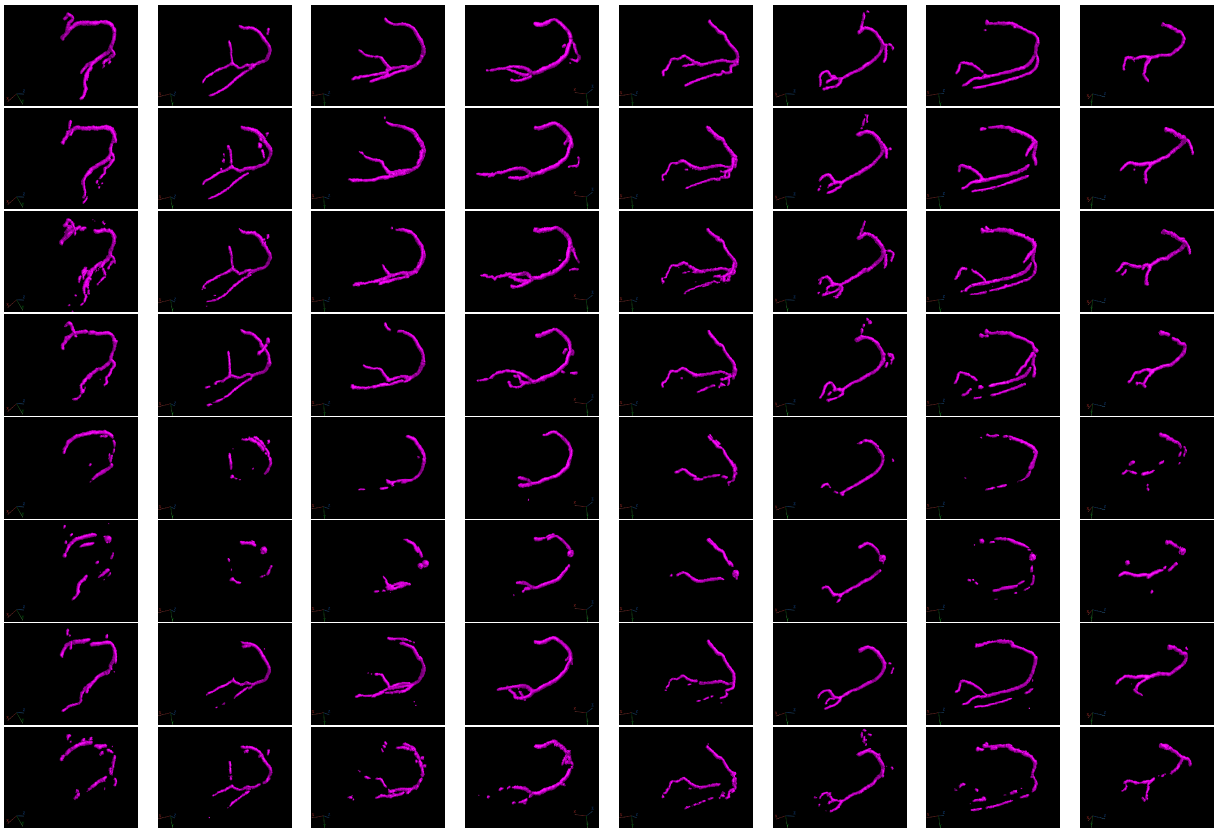


Figure B.3: 3D reconstruction results of 8 real clinical ICA data from all the models. From left to right: 8 patients. From top to bottom: 3D reconstruction results by our proposed DeepCA model, WGP, +CTLs, +DSCC, Un2+, Un3+, DSCN, and CVTG.

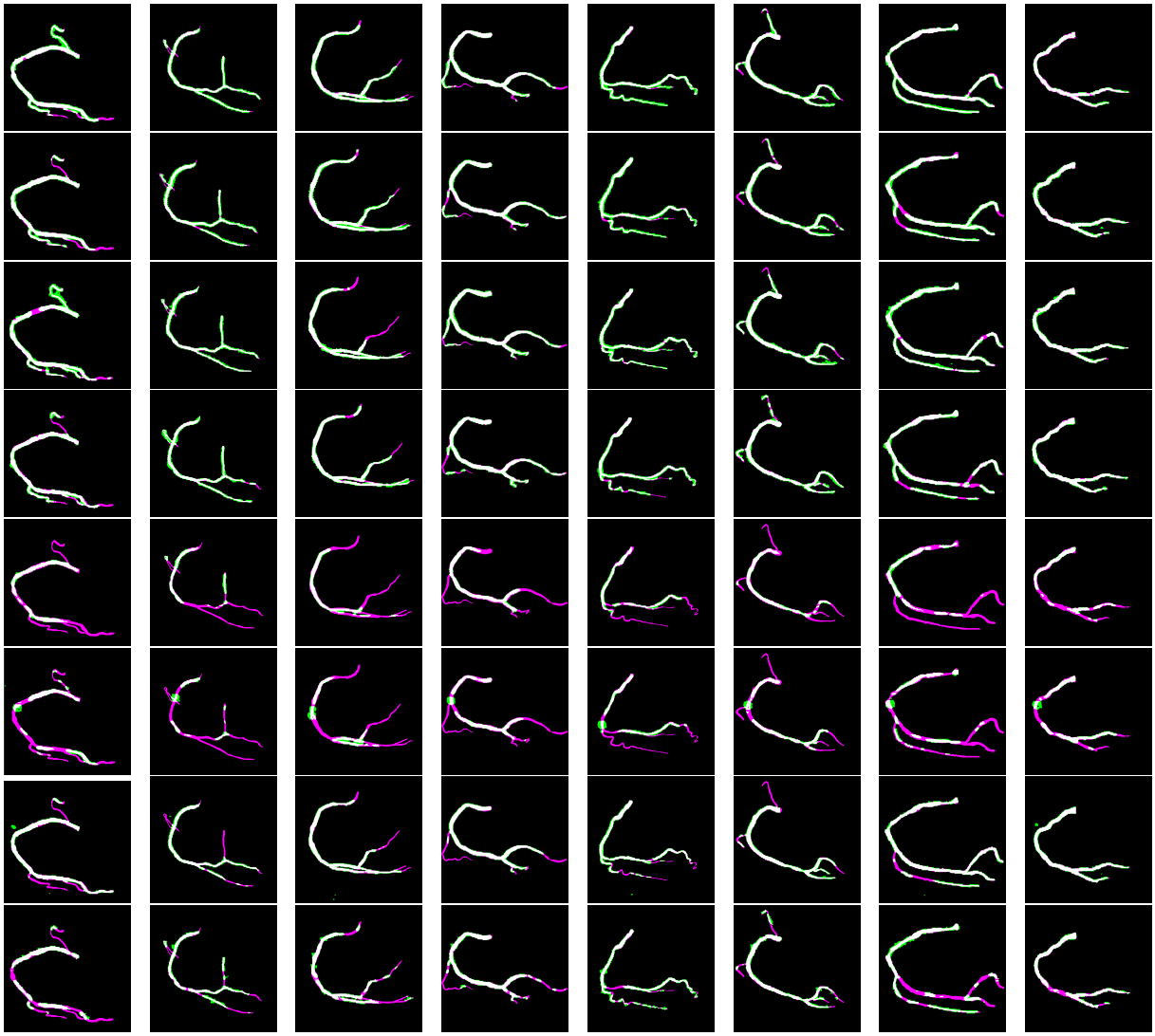


Figure B.4: Comparisons on the first projection plane between the original clinical ICA data and reprojections of the 3D reconstructions generated from all the models. From left to right: 8 patients. From top to bottom: comparisons between the original ICA data and reprojections from the reconstructions by our proposed DeepCA model, WGP, +CTLs, +DSCC, Un2+, Un3+, DSCN, and CVTG. Colour purple presents original ICA data, green is reprojection, and white shows the overlap.

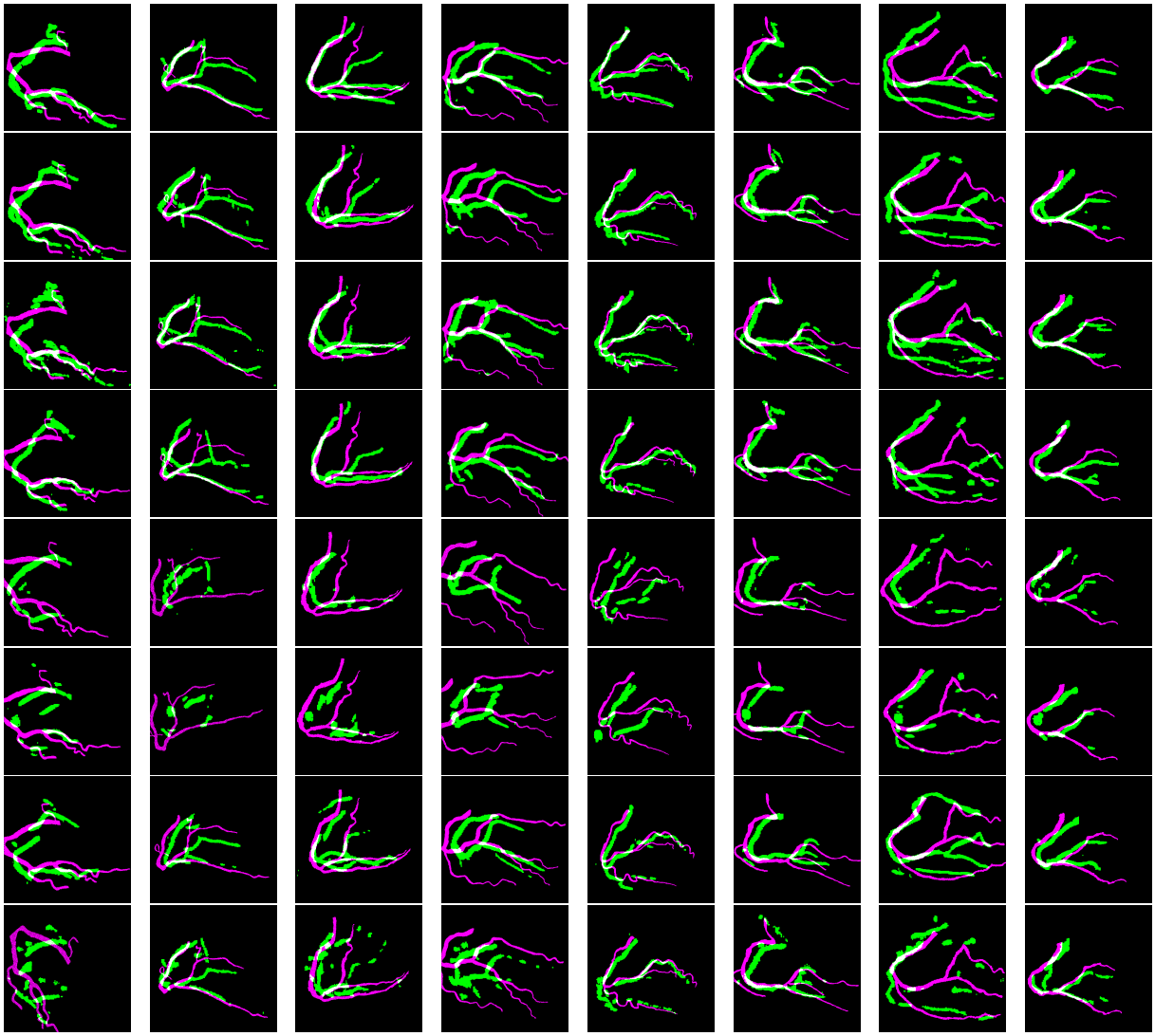


Figure B.5: Comparisons on the second projection plane between the registered ICA data and reprojections of the 3D reconstructions generated from all the models. The ICA data (in purple) are rigidly registered to the reprojections (in green) before comparison. From left to right: 8 patients. From top to bottom: comparisons between the registered ICA data and reprojections from the reconstructions by our proposed DeepCA model, WGP, +CTLs, +DSCC, Un2+, Un3+, DSCN, and CVTG. Colour purple presents registered ICA data, green is reprojection, and white shows the overlap.

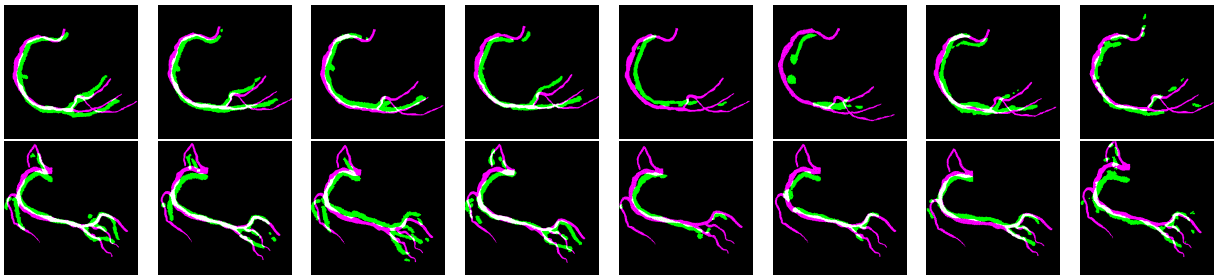


Figure B.6: Comparisons on the additional (third) projection plane between the registered ICA data and reprojections of the 3D reconstructions generated from all the models. The ICA data (in purple) are rigidly registered to the reprojections (in green) before comparison. From top to bottom: 2 patients who have additional ICA projections. From left to right: comparisons between the registered ICA data and reprojections from the reconstructions by our proposed DeepCA model, WGP, +CTLs, +DSCC, Un2+, Un3+, DSCN, and CVTG. Colour purple presents registered ICA data, green is reprojection, and white shows the overlap.

Appendix C

Iterative Motion Compensation

More Qualitative Results - Deep Iterative Motion Compensation for 3D Coronary Artery Tree Reconstruction from Two 2D Non-simultaneous Projections

In terms of CCTA, CCTA-UNSW, and VG datasets (of which CCTA-UNSW and VG datasets are from the unseen domains), for each misalignment level, we display additional reconstruction results (figures C.1, C.3 and C.6), reconstruction results combined with the corresponding ground truth in the same 3D space (figures C.4 and C.7), and the visualised corresponding voxel-wise prediction errors (mm) (figures C.2, C.5 and C.8) in terms of CD_{ℓ_2} between the ground truth and reconstruction results according to their corresponding voxel spacing, by our IterCA, DeepCAv2, and DeepCA.

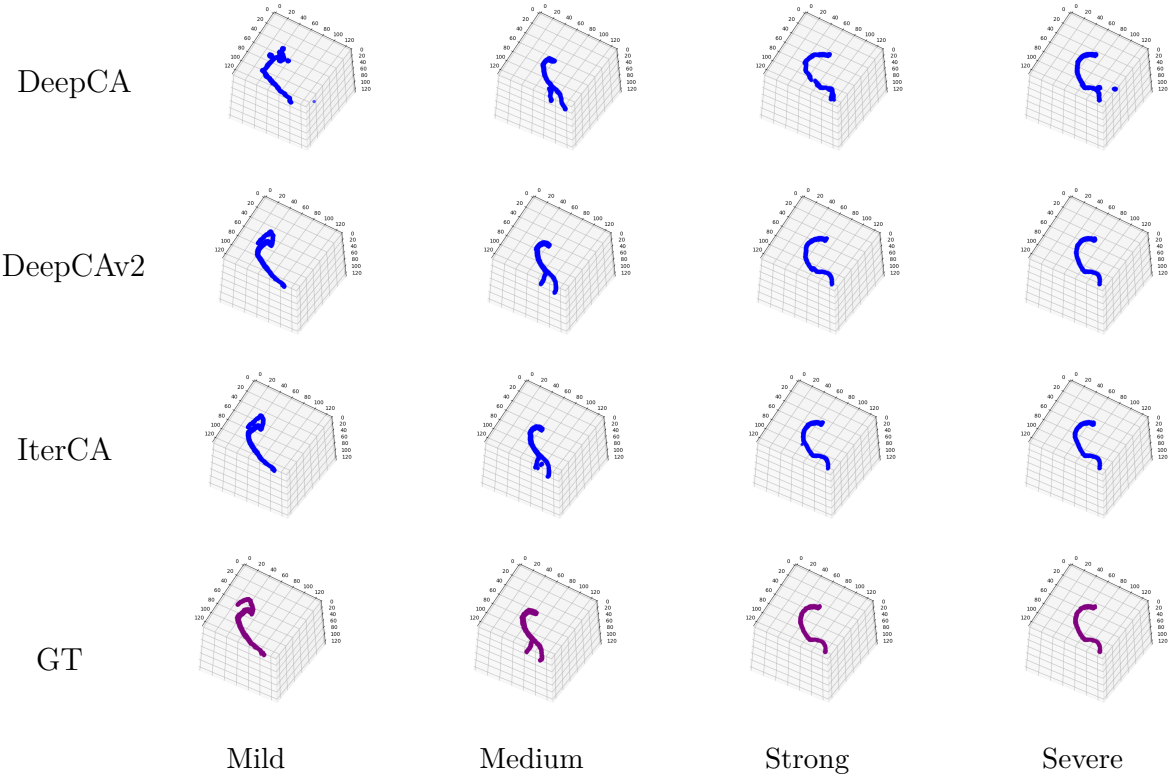


Figure C.1: Reconstruction results by our IterCA, DeepCAv2, and DeepCA (bottom to top) on four different misalignment levels (left to right) of CCTA data along with the corresponding ground truth.

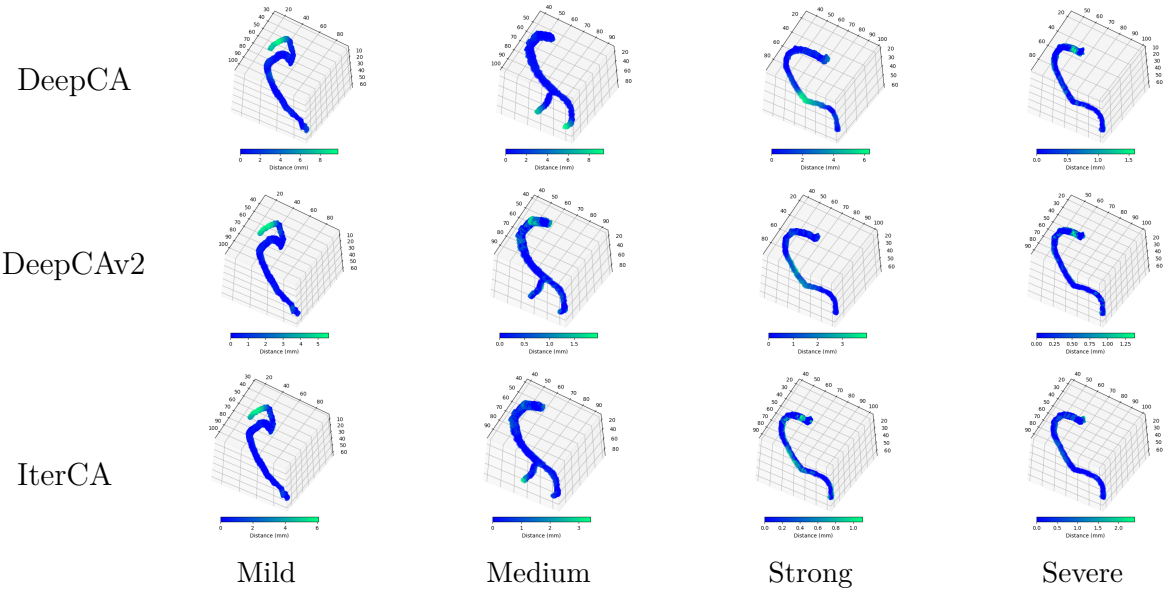


Figure C.2: Corresponding voxel-wise prediction errors (mm) in terms of CD_{ℓ_2} between the ground truth and 3D reconstruction results according to their corresponding voxel spacing. From bottom to top: visualised prediction errors by our IterCA, DeepCAv2, and DeepCA. From left to right: CCTA data with four different misalignment levels. The colour bar under each subfigure illustrates the error range for that subfigure.

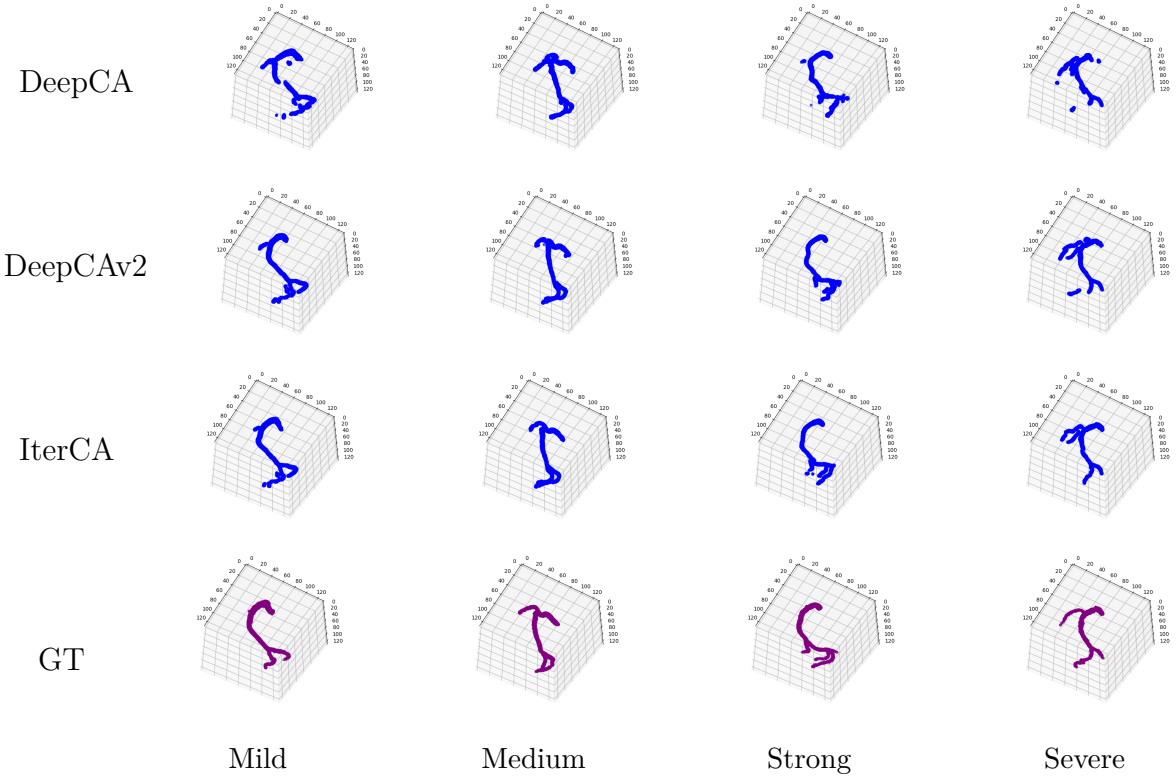


Figure C.3: Reconstruction results by our IterCA, DeepCAv2, and DeepCA (bottom to top) on four different misalignment levels (left to right) of CCTA-UNSW data along with the corresponding ground truth.

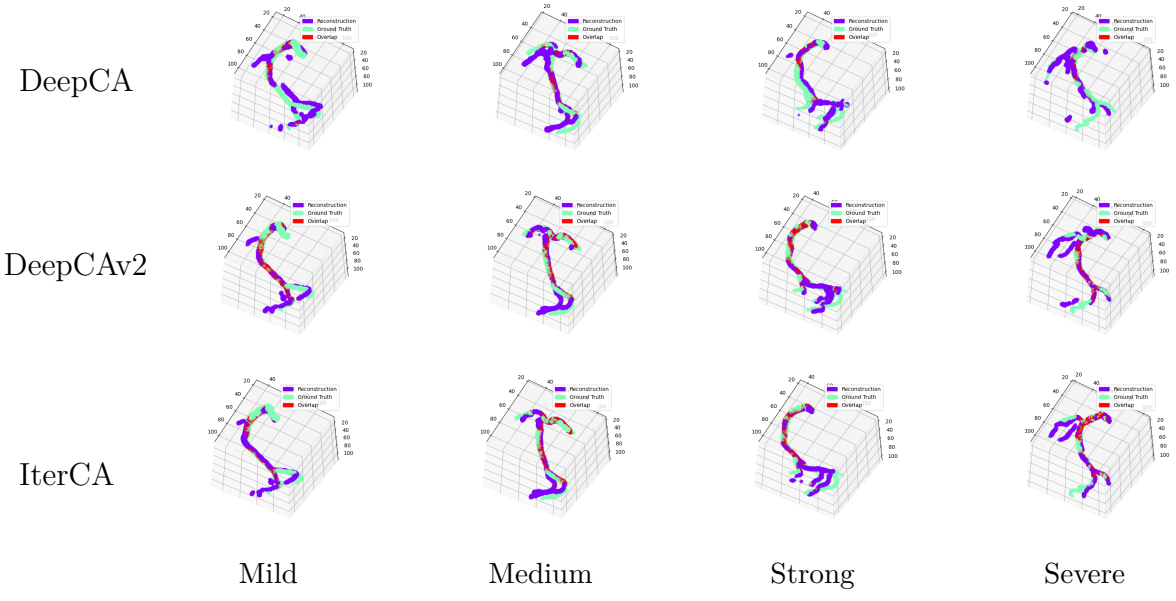


Figure C.4: Reconstruction results by our IterCA, DeepCAv2, and DeepCA (bottom to top) on four different misalignment levels (left to right) of CCTA-UNSW data combined with the corresponding ground truth in the same 3D space. Colour purple represents the reconstruction, green the ground truth, and red the overlap.

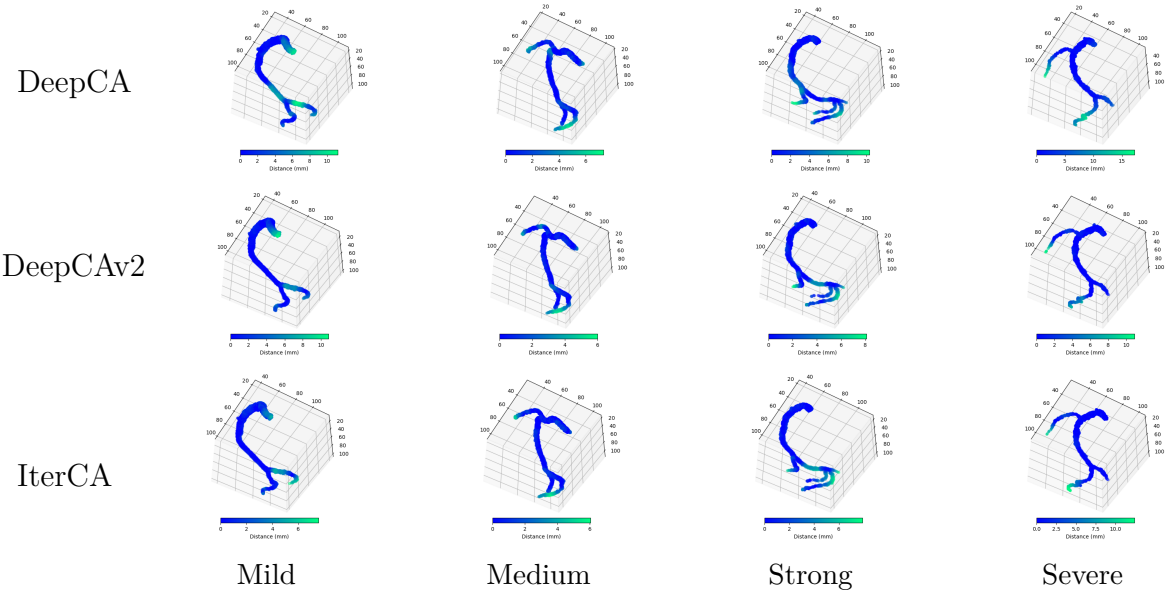


Figure C.5: Corresponding voxel-wise prediction errors (mm) in terms of CD_{ℓ_2} between the ground truth and 3D reconstruction results according to their corresponding voxel spacing. From bottom to top: visualised prediction errors by our IterCA, DeepCAv2, and DeepCA. From left to right: CCTA-UNSW data with four different misalignment levels. The colour bar under each subfigure illustrates the error range for that subfigure.

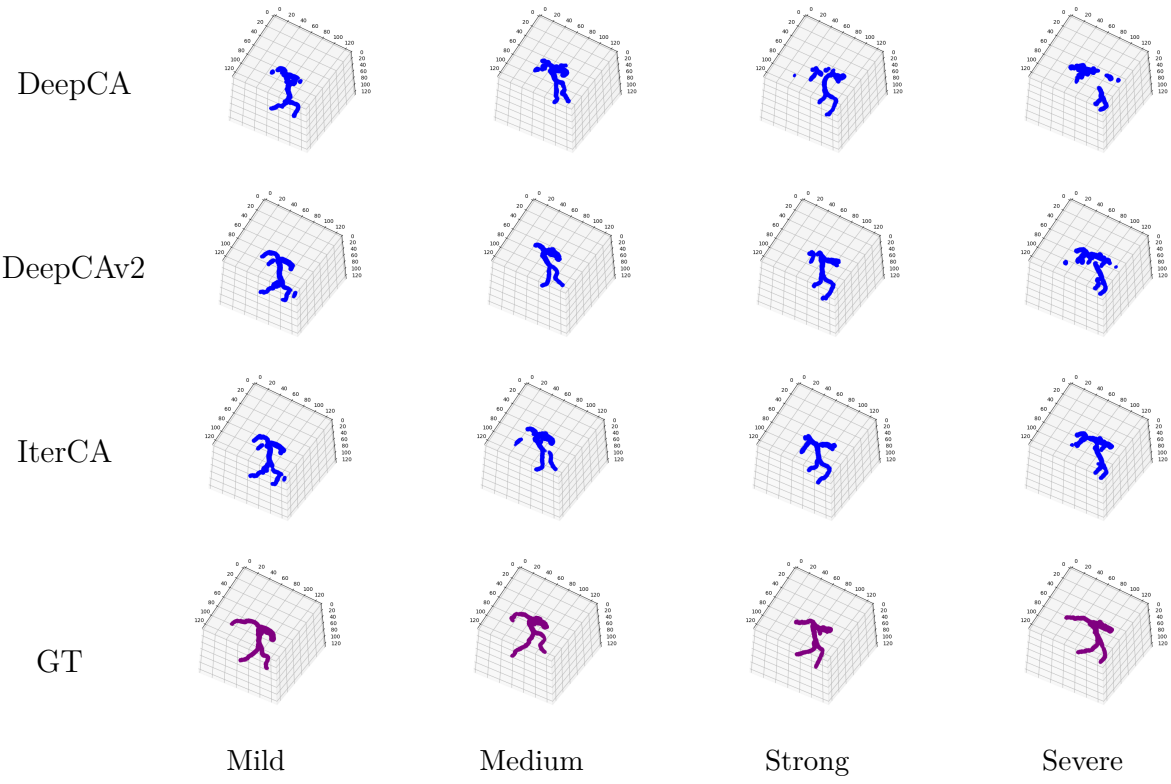


Figure C.6: Reconstruction results by our IterCA, DeepCAv2, and DeepCA (bottom to top) on four different misalignment levels (left to right) of VG data along with the corresponding ground truth.

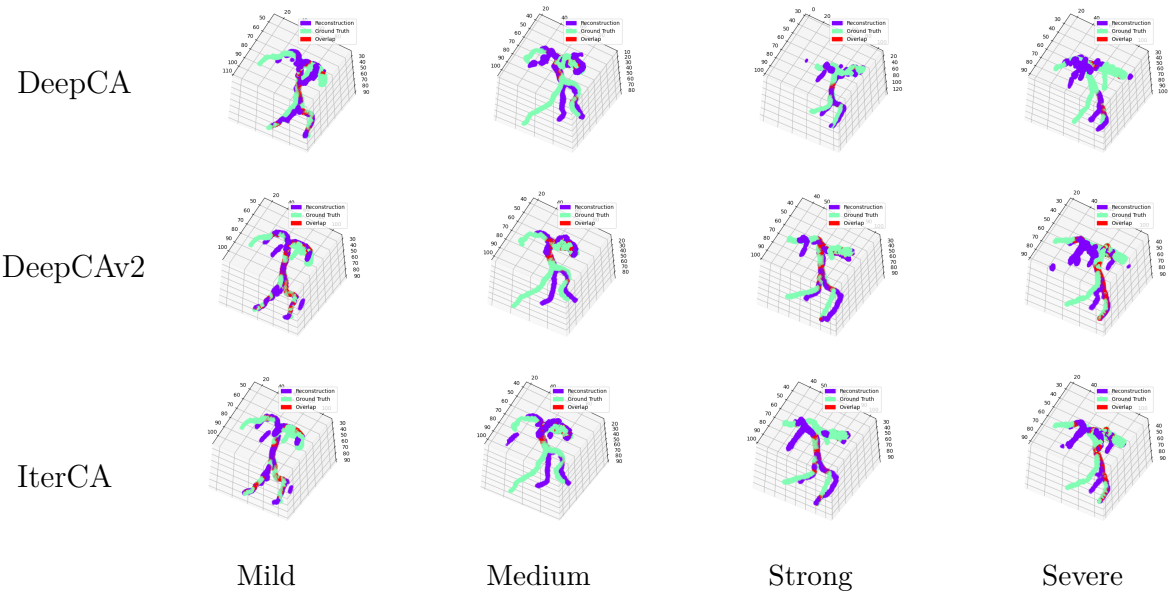


Figure C.7: Reconstruction results by our IterCA, DeepCAv2, and DeepCA (bottom to top) on four different misalignment levels (left to right) of VG data combined with the corresponding ground truth in the same 3D space. Colour purple represents the reconstruction, green the ground truth, and red the overlap.

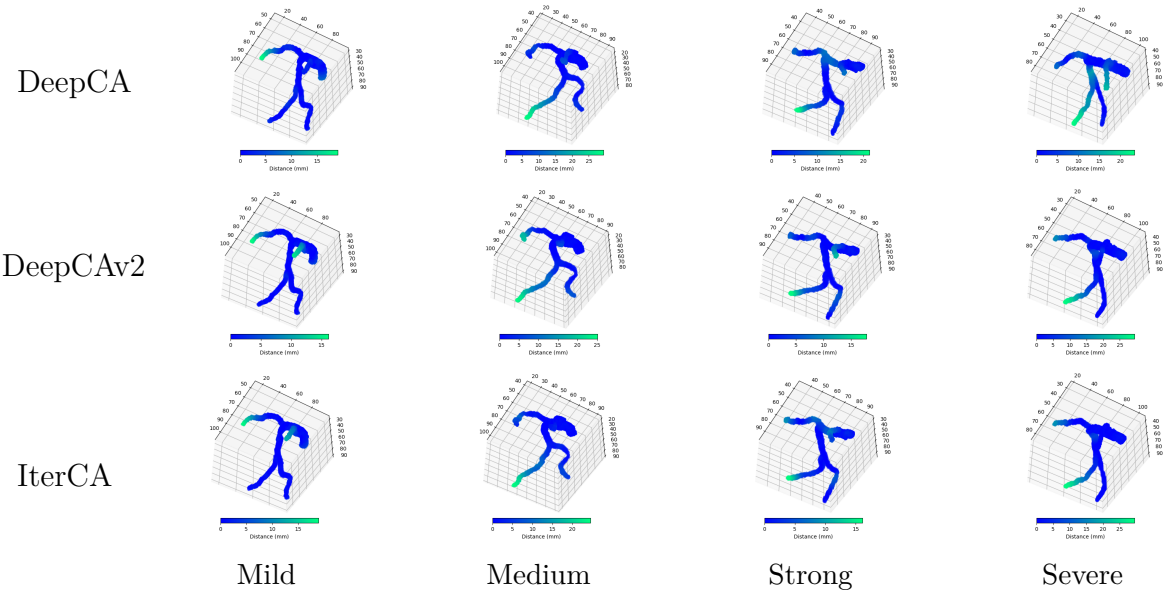


Figure C.8: Corresponding voxel-wise prediction errors (mm) in terms of CD_{ℓ_2} between the ground truth and 3D reconstruction results according to their corresponding voxel spacing. From bottom to top: visualised prediction errors by our IterCA, DeepCAv2, and DeepCA. From left to right: VG data with four different misalignment levels. The colour bar under each subfigure illustrates the error range for that subfigure.

Appendix D

Snowflake Point Transformer

More Qualitative Results - Snowflake Point Transformer with Point Adversarial Loss for Coronary Tree Reconstruction from Two Non-simultaneous Projections

We additionally present the 3D reconstruction results by our PointCA model on 8 real clinical ICA data, as illustrated in figure D.1, as well as the 3D reconstruction results along with the corresponding ground truth on three CCTA test data by our PointCA, SPT, SnowflakeNet, and PCN, as illustrated in figure D.2.

Figure D.1: Point cloud reconstruction results by our PointCA method on 8 real clinical ICA data.

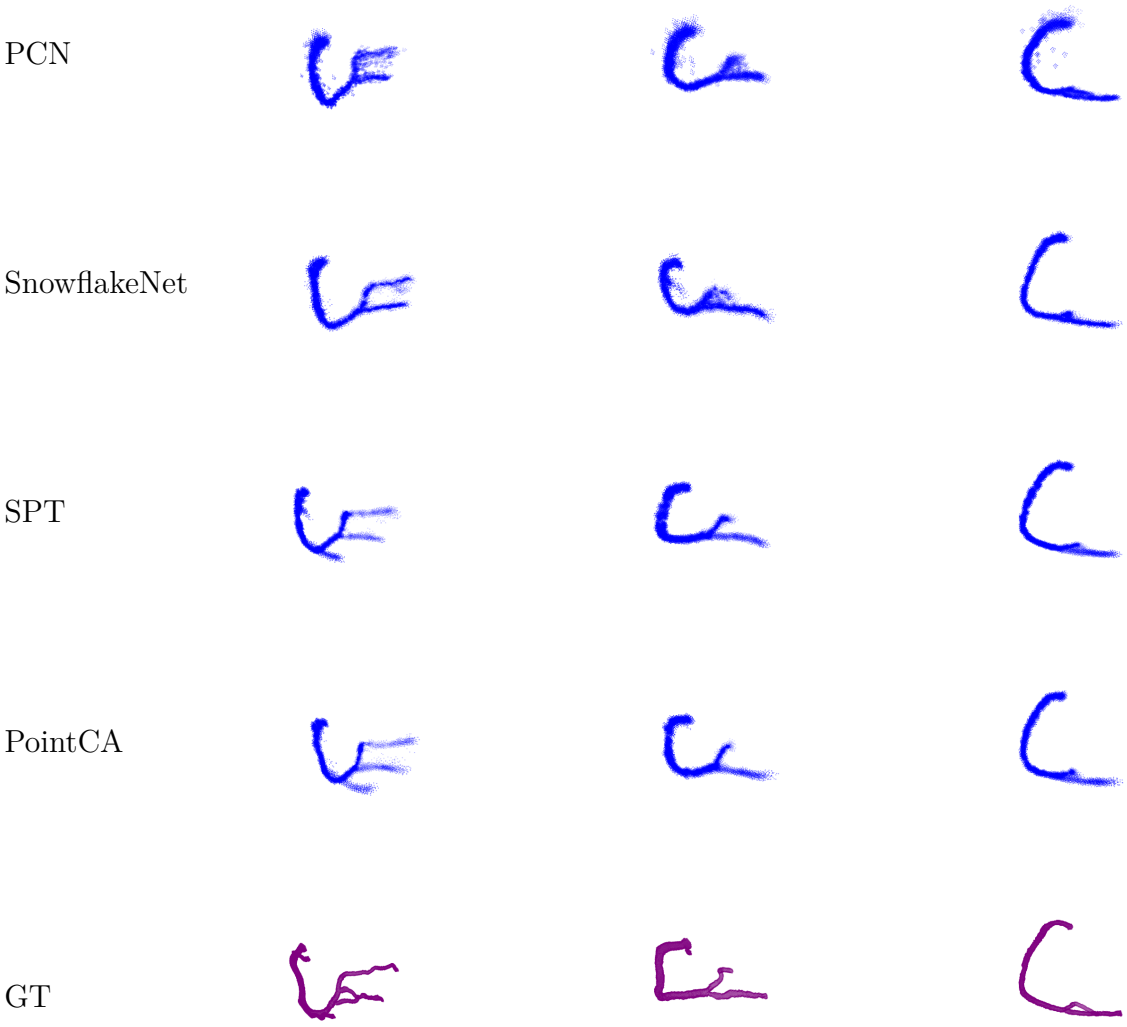


Figure D.2: Reconstruction results by our PointCA, SPT, SnowflakeNet, and PCN (bottom to top) on three CCTA test data (left to right, $R_{1,2,3}$) along with the corresponding ground truth.

Bibliography

[1] S. S. Virani, A. Alonso, H. J. Aparicio, E. J. Benjamin, M. S. Bittencourt, C. W. Callaway, A. P. Carson, A. M. Chamberlain, S. Cheng, F. N. Delling, *et al.*, "Heart disease and stroke statistics—2021 update: a report from the American Heart Association," *Circulation*, vol. 143, no. 8, pp. e254–e743, 2021.

[2] N. Townsend, D. Kazakiewicz, F. Lucy Wright, A. Timmis, R. Huculeci, A. Torbica, C. P. Gale, S. Achenbach, F. Weidinger, and P. Vardas, "Epidemiology of cardiovascular disease in Europe," *Nature Reviews Cardiology*, vol. 19, pp. 133–143, Feb. 2022.

[3] W. H. Organisation, "Cardiovascular diseases (CVDs)," 2025. `https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)`, Last accessed on 2025-07-28.

[4] S. Çimen, A. Gooya, M. Grass, and A. F. Frangi, "Reconstruction of coronary arteries from X-ray angiography: A review," *Medical image analysis*, vol. 32, pp. 46–68, 2016.

[5] B. H. Foundation, "What is an angiogram?," 2016. Accessed: 2025-07-27.

[6] N. H. S. (NHS), "Overview - Cardiac catheterisation and coronary angiography," 2022. Accessed: 2025-07-27.

[7] N. E. Green, S.-Y. J. Chen, A. R. Hansgen, J. C. Messenger, B. M. Groves, and J. D. Carroll, "Angiographic views used for percutaneous coronary interventions: A three-dimensional analysis of physician-determined vs. computer-generated views," *Catheterization and Cardiovascular Interventions*, vol. 64, no. 4, pp. 451–459, 2005.

[8] R. Smith-Bindman, P. W. Chu, H. A. Firdaus, C. Stewart, M. Malekhedayat, S. Alber, W. E. Bolch, M. Mahendra, A. B. de González, and D. L. Miglioretti,

"Projected lifetime cancer risks from current computed tomography imaging," *JAMA internal medicine*, vol. 185, no. 6, pp. 710–719, 2025.

[9] İ. ATLI, *A Novel Approach in 3D Reconstruction of Coronary Artery Tree from 2D X-ray Angiograms*. PhD thesis, Ankara Yıldırım Beyazıt Üniversitesi Fen Bilimleri Enstitüsü, 2020.

[10] B. H. Foundation, "How your heart works," 2024. Accessed: 2025-10-8.

[11] M. Unberath, *Signal processing for interventional X-ray-based coronary angiography = Signalverarbeitung für die koronarangiographie*, ch. 2.1.1, pp. 7–8. Friedrich-Alexander-Universität Erlangen-Nürnberg Technische Fakultät, 2017.

[12] H. Gray, *Anatomy of the human body*, vol. 8. Lea & Febiger, 1878.

[13] D. W. Romhilt and E. H. Estes Jr, "A point-score system for the ECG diagnosis of left ventricular hypertrophy," *American heart journal*, vol. 75, no. 6, pp. 752–758, 1968.

[14] H. G. Hosseini, D. Luo, and K. J. Reynolds, "The comparison of different feed forward neural network architectures for ECG signal diagnosis," *Medical engineering & physics*, vol. 28, no. 4, pp. 372–378, 2006.

[15] P. J. Lynch, "Coronary arteries." `https://commons.wikimedia.org/wiki/File:Coronary_arteries.svg`, 2010. Accessed 06-09-2025.

[16] "Cardiac catheterisation and coronary angiography," Oct. 2022.

[17] N. H. S. (NHS), "How they're performed - Cardiac catheterisation and coronary angiography," 2022. Accessed: 2025-07-27.

[18] J. R. Weir-McCall, M. C. Williams, A. S. Shah, G. Roditi, J. H. Rudd, D. E. Newby, and E. D. Nicol, "National trends in coronary artery disease imaging: associations with health care outcomes and costs," *Cardiovascular Imaging*, vol. 16, no. 5, pp. 659–671, 2023.

[19] N. H. S. (NHS), "Why they're used - Cardiac catheterisation and coronary angiography," 2022. Accessed: 2025-07-27.

[20] S. J. Chen and D. Schäfer, "Three-Dimensional Coronary Visualization, Part 1: Modeling," *Cardiology Clinics*, vol. 27, no. 3, pp. 433–452, 2009. Advances in Coronary Angiography.

[21] N. C. A. Programme, "National Audit of Percutaneous Coronary Intervention (NAPCI): 2025 Annual Report." `https://www.nicor.org.uk/~documents/route%3A/download/3664`, 2025. Accessed: 2025-09-05.

[22] D. Mark, D. Berman, M. Budoff, J. Carr, T. Gerber, H. Hecht, *et al.*, "Expert consensus document on coronary computed tomographic angiography," *J Am Coll Cardiol*, vol. 23, pp. 2663–2699, 2010.

[23] G. Tommasini, A. Camerini, A. Gatti, G. Derchi, A. Bruzzone, and C. Vecchio, "Panoramic Coronary Angiography," *Journal of the American College of Cardiology*, vol. 31, no. 4, pp. 871–877, 1998.

[24] A. J. Klein and J. A. Garcia, "Rotational Coronary Angiography," *Cardiology Clinics*, vol. 27, no. 3, pp. 395–405, 2009. Advances in Coronary Angiography.

[25] L. Kausch, S. Thomas, H. Kunze, M. Privalov, S. Vetter, J. Franke, A. H. Mahnken, L. Maier-Hein, and K. Maier-Hein, "Toward automatic C-arm positioning for standard projections in orthopedic surgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, pp. 1095–1105, July 2020.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, vol. 1. Cambridge: MIT press, 2016.

[27] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[28] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

[29] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[31] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[32] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, 2006.

[33] M. Ranzato, C. Poultney, S. Chopra, and Y. Cun, "Efficient learning of sparse representations with an energy-based model," *Advances in neural information processing systems*, vol. 19, 2006.

[34] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[35] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR 2015)*, Computational and Biological Learning Society, 2015.

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

[40] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[41] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in neural information processing systems*, vol. 2, 1989.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 2002.

[43] Y. LeCun, L. D. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, *et al.*, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the statistical mechanics perspective*, vol. 261, no. 276, p. 2, 1995.

[44] D. Podareanu, V. Codreanu, S. Aigner, C. v. Leeuwen, and V. Weinberg, "Best practice guide-deep learning," *Partnership for Advanced Computing in Europe (PRACE), Tech. Rep*, vol. 2, 2019.

[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[46] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[47] 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*, pp. 234–241, Springer, 2015.

[48] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[49] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017.

[50] Y. Qi, Y. He, X. Qi, Y. Zhang, and G. Yang, "Dynamic Snake Convolution Based on Topological Geometric Constraints for Tubular Structure Segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6070–6079, Oct. 2023.

[51] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR 2014)*, Computational and Biological Learning Society, 2014.

[52] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," in *International conference on machine learning*, pp. 1462–1471, Pmlr, 2015.

[53] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[54] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[55] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[56] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[57] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in GANs," in *2020 international joint conference on neural networks (ijcnn)*, pp. 1–10, Ieee, 2020.

[58] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, pp. 214–223, Pmlr, 2017.

[59] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved Training of Wasserstein GANs," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[60] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-To-Image Translation With Conditional Adversarial Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[61] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[62] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[63] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, Computational and Biological Learning Society, 2021.

[64] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[65] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

[66] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, Computational and Biological Learning Society, 2015.

[67] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2021.

[68] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[69] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

[70] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

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

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

[73] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point Transformer V3: Simpler Faster Stronger," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4840–4851, 2024.

[74] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in *2018 international conference on 3D vision (3DV)*, pp. 728–737, Ieee, 2018.

[75] P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, and Z. Han, "Snowflake Point Deconvolution for Point Cloud Completion and Generation With Skip-Transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 6320–6338, 2023.

[76] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4490–4499, 2018.

[77] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel CNN for efficient 3D deep learning," *Advances in neural information processing systems*, vol. 32, 2019.

[78] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy Networks: Learning 3D Reconstruction in Function Space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[79] W. Bian, Z. Wang, K. Li, and V. Prisacariu, "Ray-ONet: efficient 3D reconstruction from a single RGB image," in *British Machine Vision Conference 2021*, British Machine Vision Association, 2022.

[80] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 52–67, 2018.

[81] R. Tahir, A. B. Sargano, and Z. Habib, "Voxel-based 3D object reconstruction from single 2D image using variational autoencoders," *Mathematics*, vol. 9, no. 18, p. 2288, 2021.

[82] Z. Xing, Y. Chen, Z. Ling, X. Zhou, and Y. Xiang, "Few-Shot Single-View 3D Reconstruction with Memory Prior Contrastive Network," in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), pp. 55–70, Springer Nature Switzerland, 2022.

[83] K. Navaneet, A. Mathew, S. Kashyap, W.-C. Hung, V. Jampani, and R. V. Babu, "From image collections to point clouds with self-supervised shape and pose networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1132–1140, 2020.

[84] T. Hu, L. Wang, X. Xu, S. Liu, and J. Jia, "Self-supervised 3d mesh reconstruction from single images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6002–6011, 2021.

[85] X. Li, S. Liu, K. Kim, S. De Mello, V. Jampani, M.-H. Yang, and J. Kautz, "Self-supervised single-view 3d reconstruction via semantic consistency," in *European Conference on Computer Vision*, pp. 677–693, Springer, 2020.

[86] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt, *et al.*, "Wonder3d: Single image to 3d using cross-domain

diffusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9970–9980, 2024.

[87] L. Melas-Kyriazi, I. Laina, C. Rupprecht, and A. Vedaldi, "Realfusion: 360deg reconstruction of any object from a single image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8446–8455, 2023.

[88] B. Yang, S. Wang, A. Markham, and N. Trigoni, "Robust Attentional Aggregation of Deep Feature Sets for Multi-view 3D Reconstruction," *International Journal of Computer Vision*, vol. 128, pp. 53–73, Jan. 2020.

[89] J. Tang, X. Han, M. Tan, X. Tong, and K. Jia, "Skeletonnet: A topology-preserving solution for learning mesh reconstruction of object surfaces from rgb images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 10, pp. 6454–6471, 2021.

[90] J. Gwak, C. B. Choy, M. Chandraker, A. Garg, and S. Savarese, "Weakly Supervised 3D Reconstruction with Adversarial Constraint," in *2017 International Conference on 3D Vision (3DV)*, pp. 263–272, 2017.

[91] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction," in *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), pp. 628–644, Springer International Publishing, 2016.

[92] D. Jimenez Rezende, S. M. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess, "Unsupervised Learning of 3D Structure from Images," in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

[93] H. Xie, H. Yao, S. Zhang, S. Zhou, and W. Sun, "Pix2Vox++: Multi-scale Context-aware 3D Object Reconstruction from Single and Multiple Images," *International Journal of Computer Vision*, vol. 128, pp. 2919–2935, Dec. 2020.

[94] F. Wang, Q. Zhu, D. Chang, Q. Gao, J. Han, T. Zhang, R. Hartley, and M. Pollefeys, "Learning-based multi-view stereo: A survey," *arXiv preprint arXiv:2408.15235*, 2024.

[95] Z. Murez, T. Van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich, "Atlas: End-to-end 3d scene reconstruction from posed images," in *European conference on computer vision*, pp. 414–431, Springer, 2020.

[96] J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao, "NeuralRecon: Real-time coherent 3d reconstruction from monocular video," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15598–15607, 2021.

[97] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction," in *Advances in Neural Information Processing Systems: 35th conference on neural information processing systems (Neurips 2021)*, Curran Associates., 2021.

[98] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," *Advances in neural information processing systems*, vol. 34, pp. 4805–4815, 2021.

[99] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[100] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-NeRF: Point-based neural radiance fields," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5438–5448, 2022.

[101] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Transactions on Graphics*, vol. 42, July 2023.

[102] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "DUSt3R: Geometric 3D Vision Made Easy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[103] V. Leroy, Y. Cabon, and J. Revaud, "Grounding Image Matching in 3D with MASt3R," in *European Conference on Computer Vision (ECCV)*, pp. 71–91, Springer, 2024.

[104] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

[105] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3907–3916, 2018.

[106] M. Suhail, C. Esteves, L. Sigal, and A. Makadia, "Generalizable patch-based neural rendering," in *European Conference on Computer Vision*, pp. 156–174, Springer, 2022.

[107] M. Gadelha, S. Maji, and R. Wang, "3d shape induction from 2d views of multiple objects," in *2017 international conference on 3d vision (3DV)*, pp. 402–411, Ieee, 2017.

[108] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7708–7717, 2019.

[109] L. Li, S. Khan, and N. Barnes, "Silhouette-assisted 3d object instance reconstruction from a cluttered scene," in *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pp. 0–0, 2019.

[110] S. Liu, S. Saito, W. Chen, and H. Li, "Learning to infer implicit surfaces without 3d supervision," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[111] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision," *Advances in neural information processing systems*, vol. 29, 2016.

[112] Z. Han, C. Chen, Y.-S. Liu, and M. Zwicker, "DRWR: A differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images," *arXiv preprint arXiv:2007.06127*, 2020.

[113] N. I. of Health, "Computed Tomography (CT)."

[114] S. Ravishankar, J. C. Ye, and J. A. Fessler, "Image reconstruction: From sparsity to data-adaptive methods and machine learning," *Proceedings of the IEEE*, vol. 108, no. 1, pp. 86–109, 2019.

[115] G. Wang, J. C. Ye, and B. De Man, "Deep learning for tomographic image reconstruction," *Nature machine intelligence*, vol. 2, no. 12, pp. 737–748, 2020.

[116] L. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical cone-beam algorithm," *Journal of the Optical Society of America A*, vol. 1, pp. 612–619, June 1984.

[117] B. D. Smith, "Cone-beam tomography: recent advances and a tutorial review," *Optical engineering*, vol. 29, no. 5, pp. 524–534, 1990.

[118] O. Tmenova, R. Martin, and L. Duong, "CycleGAN for style transfer in X-ray angiography," *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, pp. 1785–1794, Oct. 2019.

[119] R. Martin, P. Segars, E. Samei, J. Miró, and L. Duong, "Unsupervised synthesis of realistic coronary artery X-ray angiogram," *International journal of computer assisted radiology and surgery*, vol. 18, no. 12, pp. 2329–2338, 2023.

[120] P. Henzler, V. Rasche, T. Ropinski, and T. Ritschel, "Single-image Tomography: 3D Volumes from 2D Cranial X-Rays," *Computer Graphics Forum*, vol. 37, no. 2, pp. 377–388, 2018.

[121] L. Shen, W. Zhao, and L. Xing, "Patient-specific reconstruction of volumetric computed tomography images from a single projection view via deep learning," *Nature Biomedical Engineering*, vol. 3, pp. 880–888, Nov. 2019.

[122] R. Shiode, M. Kabashima, Y. Hiasa, K. Oka, T. Murase, Y. Sato, and Y. Otake, "2D-3D reconstruction of distal forearm bone from actual X-ray images of the wrist using convolutional neural networks," *Scientific Reports*, vol. 11, p. 15249, July 2021.

[123] M. Nakao, F. Tong, M. Nakamura, and T. Matsuda, "Image-to-Graph Convolutional Network for Deformable Shape Reconstruction from a Single Projection Image," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021* (M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, eds.), pp. 259–268, Springer International Publishing, 2021.

[124] S. Yu, Y. Liu, J. Zhang, J. Xie, Y. Zheng, J. Liu, and Y. Zhao, "Cross-Domain Depth Estimation Network for 3D Vessel Reconstruction in OCT Angiography," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021* (M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, eds.), pp. 13–23, Springer International Publishing, 2021.

[125] X. Ying, H. Guo, K. Ma, J. Wu, Z. Weng, and Y. Zheng, "X2CT-GAN: reconstructing CT from biplanar x-rays with generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10619–10628, 2019.

[126] M. A. R. Ratul, K. Yuan, and W. Lee, "CCX-rayNet: A Class Conditioned Convolutional Neural Network For Biplanar x-Rays to CT Volume," in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 1655–1659, Ieee, 2021.

[127] Y. Kasten, D. Doktofsky, and I. Kovler, "End-to-end convolutional neural network for 3D reconstruction of knee bones from bi-planar x-ray images," in *International*

Workshop on Machine Learning for Medical Image Reconstruction, pp. 123–133, Springer, 2020.

[128] A. Cafaro, Q. Spinat, A. Leroy, P. Maury, A. Munoz, G. Beldjoudi, C. Robert, E. Deutsch, V. Grégoire, V. Lepetit, *et al.*, "X2Vision: 3D CT Reconstruction from Biplanar X-Rays with Deep Structure Prior," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 699–709, Springer, 2023.

[129] K. Liang, H. Yang, K. Kang, and Y. Xing, "Improve angular resolution for sparse-view CT with residual convolutional neural network," in *Medical Imaging 2018: Physics of Medical Imaging* (J. Y. Lo, T. G. Schmidt, and G.-H. Chen, eds.), vol. 10573, p. 105731k, International Society for Optics and Photonics, Spie, 2018.

[130] J. Dong, J. Fu, and Z. He, "A deep learning reconstruction framework for X-ray computed tomography with incomplete data," *Plos One*, vol. 14, pp. 1–17, Nov. 2019.

[131] L. Shen, W. Zhao, D. Capaldi, J. Pauly, and L. Xing, "A geometry-informed deep learning framework for ultra-sparse 3D tomographic image reconstruction," *Computers in Biology and Medicine*, vol. 148, p. 105710, 2022.

[132] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE transactions on image processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[133] Y. Han and J. C. Ye, "Framing U-Net via deep convolutional framelets: Application to sparse-view CT," *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1418–1429, 2018.

[134] Z. Jiang, Y. Chen, Y. Zhang, Y. Ge, F.-F. Yin, and L. Ren, "Augmentation of CBCT Reconstructed From Under-Sampled Projections Using Deep Learning," *IEEE Transactions on Medical Imaging*, vol. 38, no. 11, pp. 2705–2715, 2019.

[135] H. Xie, H. Shan, and G. Wang, "Deep Encoder-Decoder Adversarial Reconstruction (DEAR) Network for 3D CT from Few-View Data," *Bioengineering*, vol. 6, no. 4, 2019.

[136] P. Sahu, H. Huang, W. Zhao, and H. Qin, "Interactive Smoothing Parameter Optimization in DBT Reconstruction Using Deep Learning," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021* (M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, eds.), pp. 57–67, Springer International Publishing, 2021.

[137] Y. Li, K. Li, C. Zhang, J. Montoya, and G.-H. Chen, "Learning to Reconstruct Computed Tomography Images Directly From Sinogram Data Under A Variety of Data Acquisition Conditions," *IEEE Transactions on Medical Imaging*, vol. 38, no. 10, pp. 2469–2481, 2019.

[138] T. Würfl, M. Hoffmann, V. Christlein, K. Breininger, Y. Huang, M. Unberath, and A. K. Maier, "Deep Learning Computed Tomography: Learning Projection-Domain Weights From Image Domain in Limited Angle Problems," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1454–1463, 2018.

[139] M. J. Lagerwerf, D. M. Pelt, W. J. Palenstijn, and K. J. Batenburg, "A Computationally Efficient Reconstruction Algorithm for Circular Cone-Beam Computed Tomography Using Shallow Neural Networks," *Journal of Imaging*, vol. 6, no. 12, 2020.

[140] J. He, Y. Wang, and J. Ma, "Radon Inversion via Deep Learning," *IEEE Transactions on Medical Imaging*, vol. 39, no. 6, pp. 2076–2087, 2020.

[141] Y. Wang, T. Yang, and W. Huang, "Limited-Angle Computed Tomography Reconstruction using Combined FDK-Based Neural Network and U-Net," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 1572–1575, 2020.

[142] Q. Ding, H. Ji, H. Gao, and X. Zhang, "Learnable Multi-scale Fourier Interpolation for Sparse View CT Image Reconstruction," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021* (M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, eds.), pp. 286–295, Springer International Publishing, 2021.

[143] C. Wang, H. Zhang, Q. Li, K. Shang, Y. Lyu, B. Dong, and S. K. Zhou, "Improving Generalizability in Limited-Angle CT Reconstruction with Sinogram Extrapolation," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021* (M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, eds.), pp. 86–96, Springer International Publishing, 2021.

[144] R. Zha, Y. Zhang, and H. Li, "NAF: Neural Attenuation Fields for Sparse-View CBCT Reconstruction," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022* (L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li, eds.), (Cham), pp. 442–452, Springer, Springer Nature Switzerland, 2022.

[145] L. Shen, J. Pauly, and L. Xing, "NeRP: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[146] S. Park, S. Kim, I.-S. Song, and S. J. Baek, "3D Teeth Reconstruction from Panoramic Radiographs Using Neural Implicit Functions," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 376–386, Springer, 2023.

[147] C. Xu, Z. Liu, Y. Liu, Y. Dou, J. Wu, J. Wang, M. Wang, D. Shen, and Z. Cui, "TeethDreamer: 3D Teeth Reconstruction from Five Intra-oral Photographs," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 712–721, Springer, 2024.

[148] Y. Cai, Y. Liang, J. Wang, A. Wang, Y. Zhang, X. Yang, Z. Zhou, and A. Yuille, "Radiative gaussian splatting for efficient x-ray novel view synthesis," in *European Conference on Computer Vision*, pp. 283–299, Springer, 2024.

[149] R. Zha, T. J. Lin, Y. Cai, J. Cao, Y. Zhang, and H. Li, "R ^2-Gaussian: Rectifying Radiative Gaussian Splatting for Tomographic Reconstruction," *arXiv preprint arXiv:2405.20693*, 2024.

[150] K. Liang, H. Yang, and Y. Xing, "Comparison of projection domain, image domain, and comprehensive deep learning for sparse-view X-ray CT image reconstruction," *arXiv preprint arXiv:1804.04289*, 2018.

[151] G. Schoonenberg, A. Neubauer, and M. Grass, "Three-Dimensional Coronary Visualization, Part 2: 3D Reconstruction," *Cardiology Clinics*, vol. 27, no. 3, pp. 453–465, 2009. Advances in Coronary Angiography.

[152] G. Shechter, J. Resar, and E. McVeigh, "Displacement and velocity of the coronary arteries: cardiac and respiratory motion," *IEEE Transactions on Medical Imaging*, vol. 25, no. 3, pp. 369–375, 2006.

[153] L. Husmann, S. Leschka, L. Desbiolles, T. Schepis, O. Gaemperli, B. Seifert, P. Cattin, T. Frauenfelder, T. G. Flohr, B. Marincek, P. A. Kaufmann, and H. Alkadhi, "Coronary Artery Motion and Cardiac Phases: Dependency on Heart Rate-Implications for CT Image Reconstruction," *Radiology*, vol. 245, no. 2, pp. 567–576, 2007.

[154] A. Banerjee, F. Galassi, E. Zacur, G. L. De Maria, R. P. Choudhury, and V. Grau, "Point-Cloud Method for Automated 3D Coronary Tree Reconstruction From Multiple Non-Simultaneous Angiographic Projections," *IEEE Transactions on Medical Imaging*, vol. 39, no. 4, pp. 1278–1290, 2020.

[155] A. E. Holland, J. W. Goldfarb, and R. R. Edelman, "Diaphragmatic and cardiac motion during suspended breathing: preliminary experience and implications for breath-hold MR imaging.," *Radiology*, vol. 209, no. 2, pp. 483–489, 1998.

[156] F. Galassi, M. Alkhalil, R. Lee, P. Martindale, R. K. Kharbanda, K. M. Channon, V. Grau, and R. P. Choudhury, "3D reconstruction of coronary arteries from 2D angiographic projections using non-uniform rational basis splines (NURBS) for accurate modelling of coronary stenoses," *PloS One*, vol. 13, no. 1, p. e0190650, 2018.

[157] A. M. Vukicevic, S. Çimen, N. Jagic, G. Jovicic, A. F. Frangi, and N. Filipovic, "Three-dimensional reconstruction and NURBS-based structured meshing of coronary arteries from the conventional x-ray angiography projection images," *Scientific Reports*, vol. 8, no. 1, pp. 1–20, 2018.

[158] M. Unberath, O. Taubmann, A. Aichert, S. Achenbach, and A. Maier, "Prior-Free Respiratory Motion Estimation in Rotational Angiography," *IEEE Transactions on Medical Imaging*, vol. 37, no. 9, pp. 1999–2009, 2018.

[159] R. Liao, D. Luc, Y. Sun, and K. Kirchberg, "3D reconstruction of the coronary artery tree from multiple views of a rotational X-ray angiography," *The International Journal of Cardiovascular Imaging*, vol. 26, pp. 733–749, Oct. 2010.

[160] Q. Li, H. Shui, X. Hu, Y. Liu, and Y. Wang, "How to Reconstruct 3D Coronary Arterial Tree from Two Arbitrary Views," in *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*, pp. 1–4, 2009.

[161] A. Merle, G. Finet, J. Lienard, and I. Magnin, "3D reconstruction of the deformable coronary tree skeleton from two X-ray angiographic views," in *Computers in Cardiology 1998. Vol. 25 (Cat. No.98CH36292)*, pp. 757–760, 1998.

[162] T. Saito, M. Misaki, K. Shirato, and T. Takishima, "Three-dimensional quantitative coronary angiography," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 8, pp. 768–777, 1990.

[163] E. Hansis, D. Schäfer, O. Dössel, and M. Grass, "Projection-based motion compensation for gated coronary artery reconstruction from rotational X-ray angiograms," *Physics in Medicine & Biology*, vol. 53, p. 3807, June 2008.

[164] G. A. Schoonenberg, J. A. Garcia, and J. D. Carroll, "Left coronary artery thrombus characterized by a fully automatic three-dimensional gated reconstruction," *Catheterization and Cardiovascular Interventions*, vol. 74, no. 1, pp. 97–100, 2009.

[165] M. Li, H. Yang, and H. Kudo, "An accurate iterative reconstruction algorithm for sparse objects: application to 3D blood vessel reconstruction from a limited number of projections," *Physics in Medicine & Biology*, vol. 47, no. 15, p. 2599, 2002.

[166] J. Hsieh, B. Nett, Z. Yu, K. Sauer, J.-B. Thibault, and C. A. Bouman, "Recent Advances in CT Image Reconstruction," *Current Radiology Reports*, vol. 1, pp. 39–51, Mar. 2013.

[167] A. Bousse, J. Zhou, G. Yang, J.-J. Bellanger, C. Toumoulin, *et al.*, "Motion compensated tomography reconstruction of coronary arteries in rotational angiography," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 4, pp. 1254–1257, 2008.

[168] L. Wang, D.-x. Liang, X.-l. Yin, J. Qiu, Z.-y. Yang, J.-h. Xing, J.-z. Dong, and Z.-y. Ma, "Weakly-supervised 3D coronary artery reconstruction from two-view angiographic images," *arXiv preprint arXiv:2003.11846*, 2020.

[169] A. İbrahim and O. S. GEDİK, "3D reconstruction of coronary arteries using deep networks from synthetic X-ray angiogram data," *Communications Faculty of Sciences University of Ankara Series A2-A3 physical sciences and engineering*, vol. 64, no. 1, pp. 1–20, 2022.

[170] G. Y. Uluhan and Ö. Ü. O. S. Gedik, "3D Reconstruction of Coronary Artery Vessels from 2D X-Ray Angiograms and Their Pose's Details," in *2022 30th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, Ieee, 2022.

[171] K. Iyer, B. K. Nallamothu, C. A. Figueroa, and R. R. Nadakuditi, "A Multi-stage Neural Network Approach for Coronary 3D Reconstruction from Uncalibrated X-ray Angiography Images," *Scientific Reports*, 2023.

[172] K. M. Bransby, V. Tufaro, M. Cap, G. Slabaugh, C. Bourantas, and Q. Zhang, "3D Coronary Vessel Reconstruction from Bi-Plane Angiography using Graph Convolutional Networks," *arXiv preprint arXiv:2302.14795*, pp. 1–5, 2023.

[173] K. W. H. Maas, N. Pezzotti, A. J. E. Vermeer, D. Ruijters, and A. Vilanova, "NeRF for 3D Reconstruction from X-ray Angiography: Possibilities and Limitations," in *Eurographics Workshop on Visual Computing for Biology and Medicine* (C. Hansen, J. Procter, R. G. Raidou, D. Jönsson, and T. Höllt, eds.), The Eurographics Association, 2023.

[174] K. W. Maas, D. Ruijters, A. Vilanova, and N. Pezzotti, "NeRF-CA: Dynamic Reconstruction of X-ray Coronary Angiography with Extremely Sparse-views," *arXiv preprint arXiv:2408.16355*, 2024.

[175] K. W. Maas, D. Ruijters, N. Pezzotti, and A. Vilanova, "NerT-CA: Efficient Dynamic Reconstruction from Sparse-view X-ray Coronary Angiography," *arXiv preprint arXiv:2507.19328*, 2025.

[176] J. Kshirsagar, J. McNulty, B. Taji, D. So, A.-Y. Chong, P. Theriault-Lauzier, A. Wisniewski, and S. Shirmohammadi, "Generative ai-assisted novel view synthesis of coronary arteries for angiography," in *2024 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pp. 1–6, Ieee, 2024.

[177] Y. Zhu, Y. Wang, C. Di, H. Liu, F. Liao, and S. Ma, "Sparse and transferable three-dimensional dynamic vascular reconstruction for instantaneous diagnosis," *Nature Machine Intelligence*, pp. 1–13, 2025.

[178] X. Fu, Y. Li, F. Tang, J. Li, M. Zhao, G.-J. Teng, and S. K. Zhou, "3DGR-CAR: Coronary artery reconstruction from ultra-sparse 2D X-ray views with a 3D

Gaussians representation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 14–24, Springer, 2024.

[179] D. Bappy, A. Hong, E. Choi, J.-O. Park, and C.-S. Kim, "Automated three-dimensional vessel reconstruction based on deep segmentation and bi-plane angiographic projections," *Computerized Medical Imaging and Graphics*, vol. 92, p. 101956, 2021.

[180] K. Iyer, C. J. Arthurs, C. P. Najarian, R. Soroushmehr, B. K. Nallamothu, and C. A. Figueroa, "Data-Driven Approach for Coronary Vessel Reconstruction," *Circulation*, vol. 140, no. Suppl_1, pp. A12865–a12865, 2019.

[181] M. Hwang, S.-B. Hwang, H. Yu, J. Kim, D. Kim, W. Hong, A.-J. Ryu, H. Y. Cho, J. Zhang, B. K. Koo, *et al.*, "A Simple Method for Automatic 3D Reconstruction of Coronary Arteries From X-Ray Angiography," *Frontiers in Physiology*, vol. 12, p. 724216, 2021.

[182] S. M. Solutions, "Interventional Cardiac 3D software (IC3D)," 2004.

[183] Philips, "Philips's Allura 3D-CA," Jan. 2017.

[184] Medis, "Medis Suite XA."

[185] HeartFlow, "HeartFlow."

[186] J. Hsieh, E. Liu, B. Nett, J. Tang, J.-B. Thibault, and S. Sahney, "A new era of image reconstruction: TrueFidelity™," *White Paper (JB68676XX), GE Healthcare*, 2019.

[187] K. Boedeker, "AiCE deep learning reconstruction: bringing the power of ultra-high resolution CT to routine imaging," *Canon Medical Systems Corporation*, 2019.

[188] C. Arndt, F. Güttler, A. Heinrich, F. Bürckenmeyer, I. Diamantis, and U. Teichgräber, "Deep learning CT image reconstruction in clinical practice," in *RöFo-Fortschritte*

auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren, vol. 193, pp. 252–261, Georg Thieme Verlag KG, 2021.

[189] I. Wächter, *3D reconstruction of cerebral blood flow and vessel morphology from X-ray rotational angiography*. PhD thesis, UCL (University College London), 2009.

[190] H. C. Kim, B. G. Min, T. S. Lee, S. J. Lee, C. W. Lee, J. H. Park, and M. C. Han, "Three-Dimensional Digital Subtraction Angiography," *IEEE Transactions on Medical Imaging*, vol. 1, no. 2, pp. 152–158, 1982.

[191] N. Niki, Y. Kawata, H. Satoh, and T. Kumazaki, "3D imaging of blood vessels using X-ray rotational angiographic system," in *1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference*, vol. 3, pp. 1873–1877, 1993.

[192] L. Launay, P. Bouchet, E. Maurincomme, M.-O. Berger, and J.-L. Mallet, "A flexible iterative method for 3D reconstruction from X-ray projections," in *Proceedings of 13th International Conference on Pattern Recognition*, vol. 3, pp. 513–517, 1996.

[193] L. Launay, E. Maurincomme, P. Bouchet, J.-L. Mallet, and L. Picard, "3D reconstruction of cerebral vessels and pathologies from a few biplane digital angiographies," in *Visualization in Biomedical Computing* (K. H. Höhne and R. Kikinis, eds.), pp. 123–128, Springer Berlin Heidelberg, 1996.

[194] B. A. Schueler, A. Sen, H.-H. Hsiung, R. E. Latchaw, and H. Xiaoping, "Three-dimensional vascular reconstruction with a clinical X-ray angiography system," *Academic Radiology*, vol. 4, no. 10, pp. 693–699, 1997.

[195] J. Zuo, "2D to 3D Neurovascular Reconstruction from Biplane View via Deep Learning," in *2021 2nd International Conference on Computing and Data Science (CDS)*, pp. 383–387, Ieee, 2021.

[196] H. Zhao, Z. Zhou, F. Wu, D. Xiang, H. Zhao, W. Zhang, L. Li, Z. Li, J. Huang, H. Hu, *et al.*, "Self-supervised learning enables 3D digital subtraction angiography

reconstruction from ultra-sparse 2D projection views: A multicenter study," *Cell Reports Medicine*, vol. 3, no. 10, p. 100775, 2022.

[197] A. Cafaro, R. Dorent, N. Haouchine, V. Lepetit, N. Paragios, W. M. Wells III, and S. Frisken, "Two Projections Suffice for Cerebral Vascular Reconstruction," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 722–731, Springer, 2024.

[198] Z. Liu, R. Zha, H. Zhao, H. Li, and Z. Cui, "4DRGS: 4D Radiative Gaussian Splatting for Efficient 3D Vessel Reconstruction from Sparse-View Dynamic DSA Images," in *International Conference on Information Processing in Medical Imaging*, pp. 361–374, Springer, 2025.

[199] C. Otgonbaatar, J.-K. Ryu, S. Kim, J. W. Seo, H. Shim, and D. H. Hwang, "Improvement of depiction of the intracranial arteries on brain CT angiography using deep learning reconstruction," *Journal of Integrative Neuroscience*, vol. 20, no. 4, pp. 967–976, 2021.

[200] L. J. Oostveen, F. J. A. Meijer, F. de Lange, E. J. Smit, S. A. Pegge, S. C. A. Steens, M. J. van Amerongen, M. Prokop, and I. Sechopoulos, "Deep learning-based reconstruction may improve non-contrast cerebral CT imaging compared to other current reconstruction algorithms," *European Radiology*, vol. 31, pp. 5498–5506, Aug. 2021.

[201] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16560–16569, 2021.

[202] S. Çimen, M. Unberath, A. Frangi, and A. Maier, "CoronARe: A Coronary Artery Reconstruction Challenge," in *Molecular Imaging, Reconstruction and Analysis of Moving Body Organs, and Stroke Imaging and Treatment* (M. J. Cardoso, T. Arbel,

F. Gao, B. Kainz, T. van Walsum, K. Shi, K. K. Bhatia, R. Peter, T. Vercauteren, M. Reyes, A. Dalca, R. Wiest, W. Niessen, and B. J. Emmer, eds.), (Cham), pp. 96–104, Springer, Springer International Publishing, 2017.

[203] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pp. 59–66, Ieee, 1998.

[204] M. Liu, L. Sheng, S. Yang, J. Shao, and S.-M. Hu, "Morphing and sampling network for dense point cloud completion," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 11596–11603, 2020.

[205] E. Del Barrio, J. A. Cuesta-Albertos, and C. Matrán, "An optimal transportation approach for assessing almost stochastic order," in *The Mathematics of the Uncertain*, pp. 33–44, Springer, 2018.

[206] R. Dror, S. Shlomov, and R. Reichart, "Deep Dominance - How to Properly Compare Deep Neural Models," in *57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers* (A. Korhonen, D. R. Traum, and L. Màrquez, eds.), pp. 2773–2785, Association for Computational Linguistics, 2019.

[207] D. Ulmer, C. Hardmeier, and J. Frellsen, "deep-significance: Easy and Meaningful Signifcance Testing in the Age of Neural Networks," in *ML Evaluation Standards Workshop at the Tenth International Conference on Learning Representations*, 2022.

[208] N. Rauch and M. Harders, "Methods for User-Controlled Synthesis of Blood Vessel Trees in Medical Applications: A Survey," *IEEE Access*, 2025.

[209] T. Wang, Y. Lei, Y. Fu, J. F. Wynne, W. J. Curran, T. Liu, and X. Yang, "A review on medical imaging synthesis using deep learning and its clinical applications," *Journal of applied clinical medical physics*, vol. 22, no. 1, pp. 11–36, 2021.

[210] M. Zamir, "Arterial Branching within the Confines of Fractal L-System Formalism," *Journal of General Physiology*, vol. 118, pp. 267–276, Aug. 2001.

[211] X. Liu, H. Liu, A. Hao, and Q. Zhao, "Simulation of Blood Vessels for Surgery Simulators," in *2010 International Conference on Machine Vision and Human-machine Interface*, pp. 377–380, 2010.

[212] M. A. Galarreta-Valverde, M. M. G. Macedo, C. Mekkaoui, and M. P. Jackowski, "Three-dimensional synthetic blood vessel generation using stochastic L-systems," in *Medical Imaging 2013: Image Processing* (S. Ourselin and D. R. Haynor, eds.), vol. 8669, p. 86691i, International Society for Optics and Photonics, Spie, 2013.

[213] E. L. Brown, T. L. Lefebvre, P. W. Sweeney, B. J. Stolz, J. Gröhl, L. Hacker, Z. Huang, D.-L. Couturier, H. A. Harrington, H. M. Byrne, *et al.*, "Quantification of vascular networks in photoacoustic mesoscopy," *Photoacoustics*, vol. 26, p. 100357, 2022.

[214] A. Lindenmayer, "Mathematical models for cellular interactions in development I. Filaments with one-sided inputs," *Journal of Theoretical Biology*, vol. 18, no. 3, pp. 280–299, 1968.

[215] P. Prusinkiewicz and A. Lindenmayer, *The algorithmic beauty of plants*. Springer Science & Business Media, 2012.

[216] N. Rauch and M. Harders, "Interactive Synthesis of 3D Geometries of Blood Vessels," in *Eurographics 2021 - Short Papers* (H. Theisel and M. Wimmer, eds.), The Eurographics Association, 2021.

[217] G. C. Maccagnan, J. Schmith, M. Santos, and R. M. de Figueiredo, "Toolbox for vessel X-ray angiography images simulation," in *Anais do XXIII Simpósio Brasileiro de Computação Aplicada à Saúde*, pp. 59–70, Sbc, 2023.

[218] W. van Aarle, W. J. Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg, and J. Sijbers, "The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography," *Ultramicroscopy*, vol. 157, pp. 35–47, 2015.

[219] W. Van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. De Beenhouwer, K. Joost Batenburg, and J. Sijbers, "Fast and flexible X-ray tomography using the ASTRA toolbox," *Optics express*, vol. 24, no. 22, pp. 25129–25147, 2016.

[220] A. Biguri, M. Dosanjh, S. Hancock, and M. Soleimani, "TIGRE: a MATLAB-GPU toolbox for CBCT image reconstruction," *Biomedical Physics & Engineering Express*, vol. 2, no. 5, p. 055010, 2016.

[221] J. Adler, H. Kohr, and O. Öktem, "Operator Discretization Library (ODL)," Jan. 2017.

[222] A. Zeng, C. Wu, G. Lin, W. Xie, J. Hong, M. Huang, J. Zhuang, S. Bi, D. Pan, N. Ullah, K. N. Khan, T. Wang, Y. Shi, X. Li, and X. Xu, "ImageCAS: A large-scale dataset and benchmark for coronary artery segmentation based on computed tomography angiography images," *Computerized Medical Imaging and Graphics*, vol. 109, p. 102287, 2023.

[223] R. Gharleghi, D. Adikari, K. Ellenberger, S.-Y. Ooi, C. Ellis, C.-M. Chen, R. Gao, Y. He, R. Hussain, C.-Y. Lee, *et al.*, "Automated segmentation of normal and diseased coronary arteries–the ASOCA challenge," *Computerized Medical Imaging and Graphics*, vol. 97, p. 102049, 2022.

[224] R. Gharleghi, D. Adikari, K. Ellenberger, M. Webster, C. Ellis, A. Sowmya, S. Ooi, and S. Beier, "Annotated computed tomography coronary angiogram images and associated data of normal and diseased arteries," *Scientific Data*, vol. 10, no. 1, p. 128, 2023.

[225] P. J. Blanco, C. A. Bulant, L. O. Müller, G. Talou, C. G. Bezerra, P. Lemos, and R. A. Feijóo, "Comparison of 1D and 3D models for the estimation of fractional flow reserve," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[226] C. Rohkohl, G. Lauritsch, A. Keil, and J. Hornegger, "CAVAREV—an open platform for evaluating 3D and 4D cardiac vasculature reconstruction," *Physics in Medicine & Biology*, vol. 55, p. 2905, Apr. 2010.

[227] W. P. Segars, G. Sturgeon, S. Mendonca, J. Grimes, and B. M. W. Tsui, "4D XCAT phantom for multimodality imaging research," *Medical Physics*, vol. 37, no. 9, pp. 4902–4915, 2010.

[228] Y. Ma, Y. Hua, H. Deng, T. Song, H. Wang, Z. Xue, H. Cao, R. Ma, and H. Guan, "Self-supervised vessel segmentation via adversarial learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7536–7545, 2021.

[229] M. Popov, A. Amanturdieva, N. Zhaksylyk, A. Alkanov, A. Saniyazbekov, T. Aimyshev, E. Ismailov, A. Bulegenov, A. Kuzhukeyev, A. Kulanbayeva, *et al.*, "Dataset for automatic region-based coronary artery disease diagnostics using X-ray angiography images," *Scientific data*, vol. 11, no. 1, p. 20, 2024.

[230] D. Hao, S. Ding, L. Qiu, Y. Lv, B. Fei, Y. Zhu, and B. Qin, "Sequential vessel segmentation via deep channel attention network," *Neural Networks*, vol. 128, pp. 172–187, 2020.

[231] F. Cervantes-Sanchez, I. Cruz-Aceves, A. Hernandez-Aguirre, M. A. Hernandez-Gonzalez, and S. E. Solorio-Meza, "Automatic segmentation of coronary arteries in X-ray angiograms using multiscale analysis and artificial neural networks," *Applied Sciences*, vol. 9, no. 24, p. 5507, 2019.

[232] W. Silversmith, "cc3d: Connected components on multilabel 3D & 2D images.," Nov. 2021.

[233] M. Teschner, B. Heidelberger, M. Müller, D. Pomerantes, and M. H. Gross, "Optimized spatial hashing for collision detection of deformable objects," in *8th Workshop on Vision, Modeling, and Visualization (VMV)*, vol. 3, pp. 47–54, 2003.

[234] A. Maas, A. Hannun, and A. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *International Conference on Machine Learning*, (Atlanta, Georgia), 2013.

[235] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "PyTorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[236] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[237] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 424–432, Springer, 2016.

[238] H. He, A. Banerjee, M. Beetz, R. P. Choudhury, and V. Grau, "Semi-Supervised Coronary Vessels Segmentation from Invasive Coronary Angiography with Connectivity-Preserving Loss Function," in *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, Ieee, 2022.

[239] H. He, A. Banerjee, R. P. Choudhury, and V. Grau, "Automated Coronary Vessels Segmentation in X-ray Angiography Using Graph Attention Network," in *Statistical Atlases and Computational Models of the Heart. Regular and CMRxRecon Challenge Papers*, (Cham), pp. 209–219, Springer Nature Switzerland, 2024.

[240] Y. Wang, A. Banerjee, and V. Grau, "NeCA: 3D Coronary Artery Tree Reconstruction from Two 2D Projections via Neural Implicit Representation," *Bioengineering*, vol. 11, no. 12, p. 1227, 2024.

[241] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li, and H. Shi, "Escaping the Big Data Paradigm with Compact Transformers," 2022.

[242] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A Nested U-Net Architecture for Medical Image Segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 3–11, Springer, 2018.

[243] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, "UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation," 2020.

[244] P. Zeng, L. Zhou, C. Zu, X. Zeng, Z. Jiao, X. Wu, J. Zhou, D. Shen, and Y. Wang, "3D CVT-GAN: A 3D Convolutional Vision Transformer-GAN for PET Reconstruction," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pp. 516–526, 2022.

[245] Y. Wang, A. Banerjee, R. P. Choudhury, and V. Grau, "DeepCA: Deep Learning-Based 3D Coronary Artery Tree Reconstruction from Two 2D Non-Simultaneous X-Ray Angiography Projections," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 337–346, Ieee, 2025.

[246] M. Beetz, A. Banerjee, J. Ossenberg-Engels, and V. Grau, "Multi-class point cloud completion networks for 3D cardiac anatomy reconstruction from cine magnetic resonance images," *Medical Image Analysis*, vol. 90, p. 102975, 2023.

[247] H. Xu, E. Zacur, J. E. Schneider, and V. Grau, "Ventricle surface reconstruction from cardiac MR slices using deep learning," in *Functional Imaging and Modeling of*

the Heart: 10th International Conference, FIMH 2019, Bordeaux, France, June 6–8, 2019, Proceedings 10, pp. 342–351, Springer, 2019.

[248] A. G. Chandler, R. J. Pinder, T. Netsch, J. A. Schnabel, D. J. Hawkes, D. L. Hill, and R. Razavi, "Correction of misaligned slices in multi-slice cardiovascular magnetic resonance using slice-to-volume registration," *Journal of cardiovascular magnetic resonance*, vol. 10, no. 1, p. 13, 2008.

[249] G. Shechter, C. Ozturk, J. R. Resar, and E. R. McVeigh, "Respiratory motion of the heart from free breathing coronary angiograms," *IEEE transactions on medical imaging*, vol. 23, no. 8, pp. 1046–1056, 2004.

[250] K. McLeish, D. L. Hill, D. Atkinson, J. M. Blackall, and R. Razavi, "A study of the motion and deformation of the heart due to respiration," *IEEE transactions on medical imaging*, vol. 21, no. 9, pp. 1142–1150, 2002.

[251] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611, pp. 586–606, Spie, 1992.

[252] T. M. Inc., "MATLAB version: 9.13.0 (R2022b)," 2022.

[253] J.-P. Thirion, "Non-rigid matching using demons," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 245–251, IEEE, 1996.

[254] J.-P. Thirion, "Image matching as a diffusion process: an analogy with Maxwell's demons," *Medical image analysis*, vol. 2, no. 3, pp. 243–260, 1998.

[255] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pp. 876–885, Association For Uncertainty in Artificial Intelligence (AUAI), 2018.

[256] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," *arXiv preprint arXiv:1903.02428*, 2019.

[257] S. Contributors, "Spconv: Spatially Sparse Convolution Library." `https://github.com/traveller59/spconv`, 2022.

[258] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33330–33342, 2022.

[259] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *International Conference on Learning Representations*, 2019.

[260] F. D'Ascenzo, U. Barbero, E. Cerrato, M. J. Lipinski, P. Omedè, A. Montefusco, S. Taha, T. Naganuma, S. Reith, S. Voros, A. Latib, N. Gonzalo, G. Quadri, A. Colombo, G. Biondi-Zoccai, J. Escaned, C. Moretti, and F. Gaita, "Accuracy of intravascular ultrasound and optical coherence tomography in identifying functionally significant coronary stenosis according to vessel diameter: A meta-analysis of 2,581 patients and 2,807 lesions," *American Heart Journal*, vol. 169, no. 5, pp. 663–673, 2015.