PROJECTION-REFINED PARAMETRIC 3D CORONARY ARTERY RECONSTRUCTION FROM SPARSE X-RAY ANGIOGRAPHIC VIEWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse-view 3D coronary reconstruction is commonly formulated as cross-view correspondence and triangulation, which is vulnerable to vessel overlap and foreshortening, or on volumetric prediction followed by vascular-graph extraction, which does not natively provide centrelines and radii. We address both limitations with a coarse-to-fine framework that avoids explicit point matching and triangulation while directly predicting a branch-structured centreline-and-radius representation instead of an intermediate volume. From a variable number of segmented views, a coarse predictor combines frozen VGGT features with learned branch queries to estimate active branches, their B-spline centreline trajectories, and approximate dense radius profiles. Projection-guided geometry and radius refiners sample input view evidence and predict residual corrections learned with 3D supervision. We evaluate representation fidelity and sparse-view 3D reconstruction quantitatively and qualitatively. On simulated angiographic masks generated from CT-derived coronary anatomy, our method achieves the best structural and centreline metrics among the evaluated methods while maintaining competitive volumetric overlap. Coarse-to-fine inference takes 121 ms, enabling real-time reconstruction.

1 INTRODUCTION

X-ray coronary angiography is fundamental for diagnosing coronary artery disease and guiding intervention. Each angiogram, however, represents a two-dimensional projection of inherently three-dimensional coronary anatomy. Foreshortening and vessel overlap can obscure spatial relationships between branches. In clinical practice, simultaneous biplane acquisitions are uncommon, and complementary monoplane views are generally acquired sequentially. Consequently, cardiologists must mentally integrate multiple views to infer three-dimensional vessel geometry (Klein et al., 1998). Conventional 3D quantitative coronary angiography partially automates this process by reconstructing anatomy from two calibrated, phase-matched projections through cross-view vessel correspondence and triangulation (Tu et al., 2010). However, sparse projections, overlapping branches, and foreshortening can make these correspondences ambiguous. We therefore investigate whether population-level patterns of coronary anatomy learned across training cases can support a plausible branch-structured 3D hypothesis when the projection evidence is ambiguous.

Existing learning-based methods commonly represent coronary arteries using neural fields, Gaussian primitives, or voxel volumes, rather than directly producing ordered coronary centrelines and lumen radii. Extracting vascular graphs from volumetric outputs can introduce errors that compromise downstream analysis and interpretation. We instead adopt the dense centreline-and-radius representation (Iyer et al., 2023) as the model's native output. We parameterise each dense branch centreline as a continuous cubic B-spline to ensure connectivity within each branch, while retaining dense radius values for localised radius variation. This explicit representation is directly interpretable and editable, without the need to extract a vascular graph from a volumetric prediction.

We propose a coarse-to-fine model built around this artery-specific representation. Given a variable number of 2D coronary masks and their acquisition directions, the coarse predictor combines input-dependent visual information from a frozen VGGT backbone with trainable branch queries and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Comparison of learning-based coronary artery reconstruction methods. Per-case runtime is the complete time required to reconstruct one 3D artery from two input views, including vascular-graph extraction when applicable.

Method	Number of input views	Output representation	Output			Per-case runtime ↓
			3D volume	Connected centreline	Radius	
Iyer et al. (2023)	2 or 3	Dense centreline points and radii	✓	✓	✓	–
NeRF-CA (Maas et al., 2025)	≥ 4	Novel 2D views	×	×	×	–
NerT-CA (Maas et al., 2026)	≥ 3	Novel 2D views	×	×	×	–
SDF-CAR (Reda et al., 2026)	2	3D implicit SDF field	✓	×	×	45 min
3DGR-CAR (Fu et al., 2024)	≥ 2	3D Gaussian primitives	✓	×	×	63.7 s
DeepCA (Wang et al., 2025b)	2	3D voxel	✓	×	×	2.69 s
AutoCAR (Zhu et al., 2025)	2	3D voxel	✓	×	×	3.76 s
Ours	≥ 1	B-spline centreline curves and dense radii	✓	✓	✓	121 ms

prediction heads learned across training cases. It infers a plausible branch-structured 3D hypothesis: the branch-existence predictions determine the number of active branches, the B-spline control points describe their trajectories, and the dense-radius predictions provide approximate radius profiles. Subsequently, projection-guided geometry and radius refiners sample local 2D evidence from the input views and apply a fixed number of learned residual corrections to the centreline geometry and radii. This coarse-to-fine feed-forward design avoids time-consuming per-case optimisation and enables sub-second reconstruction while empirically improving accuracy. As supervised training requires paired 2D projections and 3D artery labels, we use simulated angiographic masks generated from the ImageCAS CT angiography dataset (Zeng et al., 2023) for training and evaluation. We further present exploratory qualitative examples on diseased anatomy from the ASOCA CT dataset (Gharleghi et al., 2023) to examine anatomical transfer beyond ImageCAS. Separately, we apply the preprocessing and inference pipeline to real X-ray coronary angiography to examine image-domain transfer beyond simulated inputs. We compare its reconstruction accuracy and efficiency with representative learning-based methods in Table 1.

The contributions of this work are:

- We demonstrate that integrating a coronary anatomical prior learned across the training cohort with projection-guided corrections based on case-specific evidence reconstructs accurate, anatomically structured 3D coronary arteries from sparse views, without explicit point correspondence or per-case optimisation. These results suggest that learned anatomical priors provide useful constraints under ambiguity from sparse X-ray angiographic views.
- Our coarse-to-fine framework, supported by branch-subset augmentation during training, achieves the best RCA structural and centreline reconstruction metrics among evaluated methods, with competitive volumetric accuracy and robust branch prediction. This is accomplished with sub-second inference, demonstrating both high accuracy and real-time capability.
- We enhance the explicit branch-structured centreline-and-radius representation by incorporating a smooth cubic B-spline parameterisation of dense polyline centrelines. This adaptation leads to more coherent and accurate vessel geometry reconstructions compared to dense point predictions, while maintaining interpretability and intra-branch continuity.

2 Related Work

Conventional 3D quantitative coronary angiography reconstructs coronary geometry from calibrated X-ray projections by segmenting target vessels, establishing cross-view centreline correspondence, and triangulating matched points (Klein et al., 1998; Tu et al., 2010). Related parametric approaches have used deformable B-splines for 2D vessel-boundary and centreline analysis (Klein et al., 1997), 3D B-spline curves for motion tracking in biplane angiograms (Shechter et al., 2003), and NURBS surfaces for structured coronary reconstruction and meshing (Vukicevic et al., 2018). These pipelines may require user interaction and remain sensitive to vessel overlap and foreshortening, motivating learning-based reconstruction from segmented angiograms.

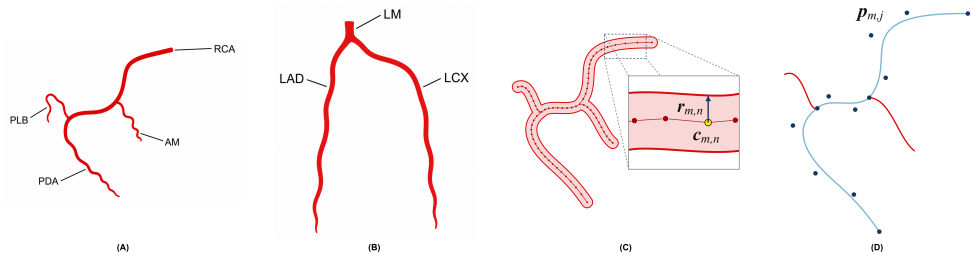


Figure 1: Coronary trees and parametric vessel representation. (A) Schematic RCA tree showing the right coronary artery (RCA), posterolateral branch (PLB), posterior descending artery (PDA), and acute marginal branch (AM); (B) Schematic LCA tree showing the left main (LM), left anterior descending (LAD), and left circumflex (LCX) arteries; (C) Dense centreline $\boldsymbol{c}_{m,n}$ and radius $r_{m,n}$ representation of a branch; (D) B-spline representation of the centreline of branch m with dark-blue points denoting B-spline control points $\boldsymbol{p}_{m,j}$ and the light-blue centreline that they define.

Recently, learning-based methods have been used for automating the mapping of angiographic images to 3D vessels. AutoCAR backprojects image features into a sparse 3D volume and predicts vascular occupancy and centreness; post-processing converts this prediction into a vascular graph, while iterative deformation relates graphs reconstructed at contracted and relaxed cardiac states (Zhu et al., 2025). DeepCA follows a related volumetric strategy: it backprojects two segmented, non-simultaneous projections into a 3D input volume and uses a conditional generative network to reconstruct a voxelised coronary tree while implicitly compensating for inter-view motion (Wang et al., 2025b). In contrast, Iyer et al. (2023) directly predict an ordered centreline-and-radius representation from segmented, uncalibrated angiograms and their model is trained and evaluated primarily on synthetically generated RCA trees with a fixed number of branches. More broadly, VesselDiffusion generates vascular point sets using a diffusion model (Guo et al., 2025). Because paired 2D projections and corresponding 3D artery labels are rarely available in clinical X-ray angiography, supervised methods commonly generate paired training data from coronary CT angiography and evaluate image-domain transfer on real X-ray angiograms.

Neural-field and Gaussian representations have also been applied to coronary reconstruction. NeRF-CA and NerT-CA model dynamic coronary appearance within an X-ray attenuation field and synthesise new composite X-ray-like views (Maas et al., 2025; 2026). 3DGR-CAR represents coronary occupancy with Gaussian primitives and combines a learned initialisation with projection-based parameter optimisation (Fu et al., 2024), whereas SDF-CAR fits a patient-specific neural implicit field to two segmented projections (Reda et al., 2026). Although these methods recover or imply 3D structure, their native neural-field, Gaussian, and occupancy representations do not directly estimate explicit coronary artery geometries for downstream tasks such as blood-flow simulation; extracting arterial structure through post-processing may introduce additional errors and does not leverage learned structural priors.

3 Methods

3.1 3D Coronary Artery Reconstruction Task

Our task is to reconstruct an explicit 3D centreline-and-radius representation of a coronary artery from a variable number of calibrated 2D coronary mask projections. For each coronary case, the input is the variable-sized set

$$\mathcal{X} = \{(\boldsymbol{I}_k, \theta_k, \phi_k)\}_{k=1}^{K}, \tag{1}$$

where K is the number of available views, $\boldsymbol{I}_k$ is the coronary artery mask in view k, and (θ_k, ϕ_k) specifies its acquisition direction. Each view follows a cone-beam projection model $\Pi_k : \mathbb{R}^3 \to \mathbb{R}^2$, whose pose is determined by (θ_k, ϕ_k) and whose fixed intrinsic parameters map 3D coordinates to detector pixels. We use the same camera model to generate the input projections and to guide the refinement described in later subsections. The target centreline-and-radius representation is introduced next.

3.2 Parametric Vessel Representation and Dataset Construction

Here, we introduce the explicit centreline-and-radius representation used as the reconstruction target and describe its construction from the training data. We refer to this representation as the *vessel code*. A complete coronary tree is represented using at most M branches, indexed by $m \in \{1, \dots, M\}$, with $e_m \in \{0, 1\}$ indicating whether branch m is present. For each existing branch, we adopt the dense centreline-and-radius representation of Iyer et al. (2023). Branch m with $e_m = 1$ is described by N ordered centreline-and-radius samples running from its proximal root or attachment point on its parent branch to its distal endpoint, as illustrated in Figure 1(C):

$$\boldsymbol{V}_m = \{(\boldsymbol{c}_{m,n}, r_{m,n})\}_{n=1}^{N}, \qquad \boldsymbol{c}_{m,n} \in \mathbb{R}^3, \quad r_{m,n} \in \mathbb{R}_{>0}. \tag{2}$$

Here, $\boldsymbol{c}_{m,n}$ denotes the n-th centreline point of branch m in 3D space, and $r_{m,n}$ denotes the vessel radius associated with that centreline point. A circular lumen profile centred at $\boldsymbol{c}_{m,n}$ with radius $r_{m,n}$ provides the corresponding tubular surface approximation.

To better reflect the continuous structure of an artery while reducing the output dimensionality, we parameterise the raw centreline coordinates by a cubic B-spline with K_{cp} control points, and the reconstruction network predicts these control points directly. For branch m, the control-point matrix $\boldsymbol{P}_m \in \mathbb{R}^{K_{\mathrm{cp}} \times 3}$ and the decoded dense centreline points are

$$\boldsymbol{P}_m = (\boldsymbol{p}_{m,1}, \dots, \boldsymbol{p}_{m,K_{\mathrm{cp}}})^{\mathsf{T}}, \qquad \boldsymbol{c}_{m,n} = \sum_{j=1}^{K_{\mathrm{cp}}} B_{j,3}(t_n)\boldsymbol{p}_{m,j}, \quad t_n = \frac{n-1}{N-1}. \tag{3}$$

Here, $\boldsymbol{p}_{m,j} \in \mathbb{R}^3$ is the j-th control point, t_n is the normalised arc-length position of centreline point n, and $B_{j,3}(\cdot)$ is the j-th cubic B-spline basis function defined by the Cox–de Boor recursion. As shown in Figure 1(D), the control points determine the decoded centreline but do not need to lie on it. Because a B-spline is a continuous curve over its parameter domain, the decoded centreline of every branch is connected by construction.

Radius is deliberately not compressed by the centreline spline. The dense vector $\boldsymbol{r}_m = (r_{m,1}, \dots, r_{m,N})^{\mathsf{T}} \in \mathbb{R}_{>0}^{N}$ retains one radius at every decoded centreline point. The complete vessel code is therefore $\{(e_m, \boldsymbol{P}_m, \boldsymbol{r}_m)\}_{m=1}^{M}$. This parameterisation provides a low-dimensional smooth prior for centreline geometry without imposing the same smoothness bottleneck on localised radius changes.

Dataset and target construction. Supervised training uses coronary annotations from the ImageCAS CT angiography dataset (Zeng et al., 2023) as 3D labels. Each annotation is separated into RCA and LCA components (Figure 1(A–B)). Each component mask is skeletonised to extract ordered branch centrelines, and the radius at each centreline point is estimated using the Euclidean distance transform, obtaining an ordered branch-wise vessel code. Branches follow deterministic appearance-based positions in the target sequence, allowing optional side branches without assigning every query a fixed anatomical name. Each retained branch contains $N = 200$ proximal-to-distal centreline-and-radius samples, represented by $K_{\mathrm{cp}} = 20$ B-spline control points and dense radii in millimetres. This process yields 750 RCA and 750 LCA components, with each cohort divided into 600 training, 75 validation, and 75 test cases. For every ImageCAS case, the cone-beam model Π_k generates seven 256×256 synthetic coronary-mask projections that mimic segmented X-ray angiograms at clinically motivated acquisition directions, from which the requested number of views is selected during training and evaluation. Separately, the 20 diseased CT angiography cases from ASOCA (Gharleghi et al., 2023) are used only for qualitative analysis; their annotations are not converted into vessel code targets or included in training or quantitative evaluation.

3.3 Flexible-View Coarse Parametric Predictor

The first stage of our model is the flexible-view coarse parametric predictor summarised in Figure 2. The coarse predictor combines evidence from the available views with learned patterns of coronary anatomy to infer a plausible structured vessel code. The existence predictions determine the active number of branches, while the control-point and radius heads estimate their global trajectories and approximate radius profiles.

As illustrated in Figure 2(A), the frozen VGGT image backbone f_{backbone} extracts image tokens $\boldsymbol{H}_k^I$ from each input image $\boldsymbol{I}_k$, while the view-direction encoder f_{view} encodes its acquisition direction

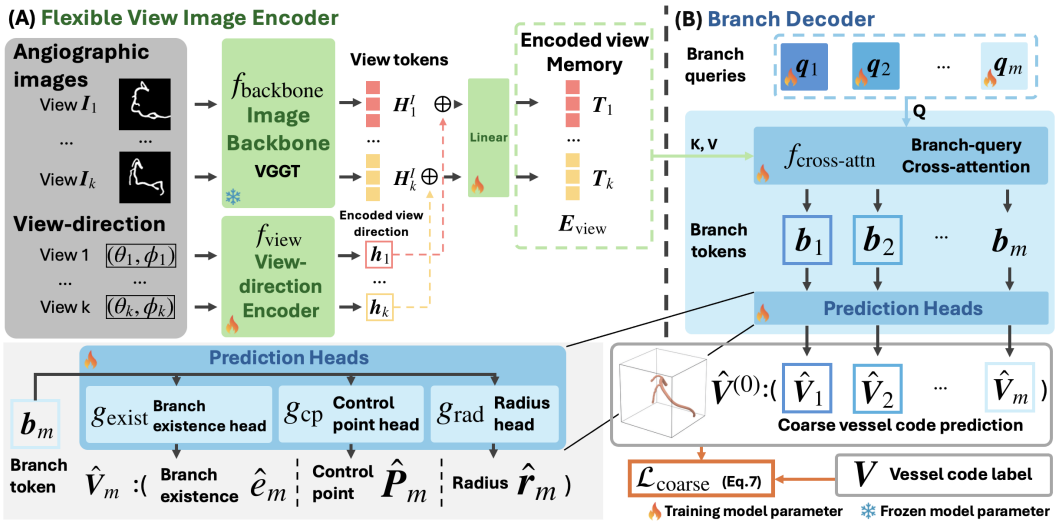


Figure 2: Flexible-view coarse parametric predictor: (A) Frozen VGGT features are concatenated ($\oplus$) with encoded acquisition directions to form the multiview memory $\boldsymbol{E}_{\text{view}}$. (B) Trainable branch queries cross-attend to this memory, and the existence, control-point, and radius heads produce the coarse vessel code $\hat{\boldsymbol{V}}^{(0)}$. Fire and snowflake symbols denote trainable and frozen modules, respectively.

(θ_k, ϕ_k) as $\boldsymbol{h}_k$. To obtain the view-direction-aware tokens $\boldsymbol{T}_k$, the same vector $\boldsymbol{h}_k$ is repeated for every image token, concatenated with $\boldsymbol{H}_k^I$ along the feature dimension, and passed through a learned projection back to the model dimension d. The tokens from all available views form the multiview memory $\boldsymbol{E}_{\text{view}} = [\boldsymbol{T}_1; \ldots; \boldsymbol{T}_K]$. The number of views K and their acquisition directions can vary between cases.

Figure 2(B) shows how M trainable branch queries $\boldsymbol{q}_m$ retrieve branch-specific evidence from this memory. The query matrix $\boldsymbol{Q}_{\text{query}} = [\boldsymbol{q}_1; \ldots; \boldsymbol{q}_M]$ cross-attends to $\boldsymbol{E}_{\text{view}}$ through $f_{\text{cross-attn}}$, with $\boldsymbol{Q}_{\text{query}}$ serving as the queries and $\boldsymbol{E}_{\text{view}}$ as the keys and values. This produces branch tokens $[\boldsymbol{b}_1; \ldots; \boldsymbol{b}_M]$, where $\boldsymbol{b}_m \in \mathbb{R}^d$. The branch queries correspond to the ordered branch entries: $\boldsymbol{q}_m$ retrieves image evidence for the m-th branch entry, while the existence head determines whether that branch is present in the current case. The resulting branch token $\boldsymbol{b}_m$ is mapped by the existence, control-point, and radius heads to the coarse vessel code $\hat{\boldsymbol{V}}^{(0)} = \left\{(\hat{e}_m, \hat{\boldsymbol{P}}_m^{(0)}, \hat{\boldsymbol{r}}_m^{(0)})\right\}_{m=1}^{M}$, where $\hat{e}_m$, $\hat{\boldsymbol{P}}_m^{(0)}$, and $\hat{\boldsymbol{r}}_m^{(0)}$ denote the predicted branch existence, B-spline control points, and dense radii, respectively. This stage learns the mapping from input appearance to 3D artery structure through paired training data.It therefore produces a plausible prior-guided 3D hypothesis but does not enforce direct projection-space consistency. The following refinement stage uses input view evidence to correct this hypothesis.

3.4 Projection-Guided Refinement

The second part of the model, summarised in Figure 3, progressively refines coarse vessel code $\hat{\boldsymbol{V}}^{(0)}$. At each refinement stage, the current centreline and radii are projected into every available input view using the same cone-beam operator Π_k employed to generate the input projections. Local 2D evidence is sampled around these projected positions and aggregated across views through cross-attention. A residual head then predicts a correction to the current 3D parameter.

For the geometry refiner, at refinement stage s, the query associated with control point $\hat{\boldsymbol{p}}_{m,j}^{(s)}$ cross-attends to the multiview evidence sampled around its corresponding projected centreline position. The resulting geometry residual $\Delta\hat{\boldsymbol{p}}_{m,j}^{(s)}$ updates the control point as $\hat{\boldsymbol{p}}_{m,j}^{(s+1)} = \hat{\boldsymbol{p}}_{m,j}^{(s)} + \Delta\hat{\boldsymbol{p}}_{m,j}^{(s)}$. The updated B-spline is decoded and reprojected before the next stage samples new evidence. We use $S_g = 4$ shared-parameter geometry-refinement stages while holding the branch-existence predic-

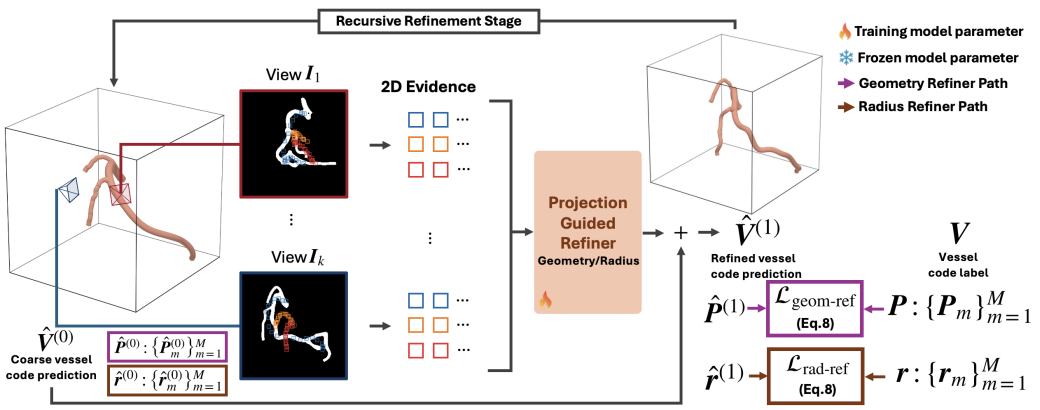


Figure 3: Recursive projection-guided refinement. The B-spline geometry refiner uses local multi-view evidence to predict residual updates to the coarse control points. With the refined centreline fixed, the dense-radius refiner then updates the pointwise radii. The two refiners are trained sequentially and separately, using $\mathcal{L}_{\text{geom-ref}}$ and $\mathcal{L}_{\text{rad-ref}}$, respectively.

tions and coarse radii fixed. After geometry refinement, the dense-radius refiner applies the same residual principle at every decoded centreline point. A radius query cross-attends to the local multi-view evidence and predicts $\Delta\hat{r}_{m,n}^{(s)}$, giving $\hat{r}_{m,n}^{(s+1)} = \hat{r}_{m,n}^{(s)} + \Delta\hat{r}_{m,n}^{(s)}$. The refined centreline geometry is held fixed throughout this module. We use $S_r = 3$ shared-parameter radius-refinement stages, producing the final vessel code $\hat{V}_{\text{refined}}$.

3.5 Staged Training Objectives

The model is trained sequentially: first the coarse predictor, then the geometry refiner with the coarse predictor frozen, and finally the radius refiner with the preceding modules frozen. Because target branches have a deterministic ordering, prediction m is compared directly with ground-truth branch m. At prediction stage s, the geometry loss supervises both the B-spline control points and the centreline decoded from them:

$$\mathcal{L}_{\text{geometry}}^{(s)} = \frac{\lambda_{\text{cp}}}{N_{\text{cp}}} \sum_{m,j} e_m \left\| \hat{\boldsymbol{p}}_{m,j}^{(s)} - \boldsymbol{p}_{m,j} \right\|_2^2 + \frac{\lambda_{\text{cl}}}{N_{\text{cl}}} \sum_{m,n} e_m \left\| \hat{\boldsymbol{c}}_{m,n}^{(s)} - \boldsymbol{c}_{m,n} \right\|_2^2, \tag{4}$$

where N_{cp} and N_{cl} denote the total numbers of control points and decoded centreline points across existing branches, and λ_{cp} and λ_{cl} weight the corresponding supervision terms. While the B-spline parameterisation guarantees intra-branch connectivity, the attachment loss encourages connectivity between side branches and their parent branch. Let $\kappa_m \in \{1, \ldots, N\}$ denote the ground-truth index of this junction on the parent branch centreline. We regularise the predicted junction by

$$\mathcal{L}_{\text{attach}}^{(s)} = \frac{1}{\sum_m e_m} \sum_m e_m \left\| \hat{\boldsymbol{c}}_{m,1}^{(s)} - \hat{\boldsymbol{c}}_{p,\kappa_m}^{(s)} \right\|_2^2, \tag{5}$$

where p denotes the index of the parent branch. This loss directly encourages the predicted first point of each existing side branch to coincide with its corresponding predicted parent branch attachment point. The radius loss is the pointwise mean absolute error at corresponding dense centreline points:

$$\mathcal{L}_{\text{radius}}^{(s)} = \frac{1}{N_{\text{cl}}} \sum_{m,n} e_m \left| \hat{r}_{m,n}^{(s)} - r_{m,n} \right|. \tag{6}$$

Across these losses, e_m excludes absent branches. For the coarse prediction, $s = 0$, and binary cross-entropy $\mathcal{L}_{\text{exist}}$ supervises the predicted branch-existence labels:

$$\mathcal{L}_{\text{coarse}} = \mathcal{L}_{\text{geometry}}^{(0)} + \lambda_{\text{attach}} \mathcal{L}_{\text{attach}}^{(0)} + \lambda_{\text{radius}} \mathcal{L}_{\text{radius}}^{(0)} + \lambda_{\text{exist}} \mathcal{L}_{\text{exist}}, \tag{7}$$

Absent branch entries therefore contribute to existence supervision but not to geometry, attachment, or radius supervision. For refinement, the losses used are

$$\mathcal{L}_{\text{geom-ref}} = \sum_{s=1}^{S_g} \alpha_s (\mathcal{L}_{\text{geometry}}^{(s)} + \lambda_{\text{attach}} \mathcal{L}_{\text{attach}}^{(s)}), \qquad \mathcal{L}_{\text{rad-ref}} = \sum_{s=1}^{S_r} \beta_s \mathcal{L}_{\text{radius}}^{(s)}, \tag{8}$$

where $\mathcal{L}_{\text{geom-ref}}$ collects geometry and attachment losses over the recursive geometry stages, while $\mathcal{L}_{\text{rad-ref}}$ collects radius losses over the radius stages. Here α_s and β_s are loss weights. These objectives are applied separately: $\mathcal{L}_{\text{geom-ref}}$ trains only the geometry refiner, whereas $\mathcal{L}_{\text{rad-ref}}$ trains only the radius refiner. The same existence and valid-point masks are applied throughout refinement.

Within-case branch-subset augmentation training. To model optional side branches, each training tree is paired with variants formed by successively removing side branches in reverse canonical order until only the required main path remains. Updated projections, existence labels, and geometry targets from all variants train the coarse predictor and geometry refiner, varying branch presence while retaining the same underlying anatomy.

3.6 Evaluation Metrics

Reconstruction is evaluated on a common 0.5 mm isotropic grid using Dice for volumetric overlap, clDice for topology-aware centreline agreement, symmetric Chamfer distance (CD) for centreline localisation, and connected-component (CC) count for fragmentation, whose ideal value is one. We compute each metric per case and report its mean and standard error across cases; higher Dice and clDice and lower CD and component count indicate better reconstruction.

Additional data-construction and architectural details are provided in Appendix Sections A.1 and A.2, respectively.

4 Results

4.1 Comparison model selection

We quantitatively compare against 3D reconstruction methods 3DGR-CAR (Fu et al., 2024), AutoCAR (Zhu et al., 2025), and DeepCA (Wang et al., 2025b); the latter two are evaluated only in their fixed two-view setting. NeRF-CA (Maas et al., 2025) and NerT-CA (Maas et al., 2026) generate novel X-ray-like views rather than 3D arteries; the method of Iyer et al. (2023) operates on uncalibrated input views and assumes a fixed number of artery branches; therefore these methods are not evaluated. SDF-CAR reconstructs a 3D artery without cross-case pre-training, but its per-patient optimisation requires approximately 45 minutes per case (Reda et al., 2026), and thus is excluded from quantitative comparison.

4.2 Main Results

Table 2 evaluates reconstruction from one, two, and four input views on the 75-case RCA test split described in Section 3.2, using Dice, clDice, Chamfer distance, and connected-component count. For methods without a native centreline graph, we extract the centreline using the post-processing approach of Zhu et al. (2025). All methods use the same view subsets and evaluation protocol.

Compared with the other methods in Table 2, our refined model outperforms 3DGR-CAR in the one-view setting. Under the conventional two-view setting, AutoCAR achieves the highest Dice, whereas our method achieves the highest clDice and lowest Chamfer distance, indicating stronger centreline-topology agreement and localisation despite the small difference in volumetric overlap. With four views, our refined model substantially outperforms 3DGR-CAR across Dice, clDice, and Chamfer distance. Dice alone does not fully capture vessel connectivity because disconnected foreground regions may still overlap with the target artery. As each branch centreline is connected by construction and attachment supervision encourages side branches to join the main branch, our refined predictions achieve the lowest CC count at every evaluated view setting and remain close to the ideal value of one, while the comparison methods produce more fragmented outputs. On the more complex LCA cohort (Appendix Table 9), our method achieves the lowest connected-component count (1.05 versus 9.97 for the next-best method), indicating a substantially more coherent artery-tree structure.

Our model ablations clarify the contribution of each design choice. The difference between the coarse and refined results illustrates their complementary roles: the coarse predictor establishes the branch structure and broad artery geometry, whereas projection-guided refinement introduces

Table 2: RCA reconstruction from one, two, and four input views. CD denotes Chamfer distance, and CC count denotes connected-component count. Values are mean ± standard error. The best result is shown in **bold** and the second-best result is underlined. – denotes a view-count setting that is not reported in the corresponding paper.

Method	One view				Two views				Four views			
	Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓	Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓	Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓
3DGR-CAR (Fu et al., 2024)	22.18 (±0.65)	22.35 (±0.73)	10.11 (±0.44)	72.77 (±12.08)	37.32 (±0.91)	39.47 (±1.09)	7.72 (±0.48)	159.62 (±16.84)	44.33 (±1.12)	46.54 (±1.31)	7.37 (±0.49)	216.25 (±17.95)
AutoCAR (Zhu et al., 2025)	–	–	–	–	**68.76** (±0.70)	79.13 (±1.37)	3.24 (±0.27)	3.35 (±0.31)	–	–	–	–
DeepCA (Wang et al., 2025b)	–	–	–	–	63.49 (±1.56)	78.00 (±1.73)	3.78 (±0.51)	4.41 (±0.60)	–	–	–	–
Ours, coarse	15.73 (±1.46)	16.43 (±1.84)	11.86 (±0.59)	1.324 (±0.072)	19.47 (±1.12)	20.40 (±1.56)	9.70 (±0.41)	1.324 (±0.064)	21.45 (±1.26)	22.88 (±1.70)	9.16 (±0.36)	1.284 (±0.059)
Ours, refined	**27.37** (±1.86)	**32.38** (±2.56)	**9.39** (±0.55)	**1.081** (±0.032)	66.92 (±1.50)	**80.62** (±1.92)	**3.16** (±0.27)	**1.054** (±0.026)	**73.05** (±1.25)	**88.54** (±1.45)	**2.12** (±0.19)	**1.014** (±0.014)

case-specific corrections from the input evidence. Replacing direct dense centreline prediction with B-spline control-point prediction increases 2-view RCA refined Dice from 22.18% to 66.92% (Appendix Table 5). Branch-subset augmentation training increases geometry-refined Dice from 46.57% to 65.89%, branch-existence accuracy from 92.95% to 94.63% and F1 from 88.37% to 92.48%, indicating more reliable identification of optional side branches (Appendix Table 7). Starting from the same coarse prediction, the learned geometry refiner achieves 65.89% Dice compared with 41.02% for per-case optimisation (Appendix Table 10), suggesting that 3D-supervised refinement can use projection evidence more reliably than directly fitting sparse 2D observations.

Figure 4 qualitatively illustrates these connectivity differences by showing gaps in voxel and implicit predictions. Subsequent graph extraction may fail to fill the gaps in the predicted artery. In contrast, our continuous branch representation preserves within-branch connectivity and exposes branch attachments and radii. The arrows in Figure 4 mark branches that are weakly visible in the inputs but recovered by our model, suggesting that learned anatomical patterns complement sparse projection evidence. Figure 5(A) shows that the coarse prediction captures the global artery shape, while projection-guided refinement uses 2D evidence to recover the corresponding radius reduction. This indicates sensitivity to visible narrowing, rather than clinical stenosis detection. Figure 5(B) demonstrates application to preprocessed real angiography, although the absence of paired 3D labels prevents accuracy assessment. Additional qualitative video results are provided in the supplementary video.

Implementation details and comparison-method adaptations are described in Appendix Sections A.3.1 and A.3.3, respectively. An additional ablation of the frozen image backbone and an inference-time breakdown for the comparison methods are reported in Appendix Tables 6 and 8.

5 Discussion and Conclusion

We propose a coarse-to-fine model that combines anatomical patterns learned across the training cohort with projection-guided corrections from the input views. Its explicit vessel-code representation guarantees a connected centreline within each branch, while attachment supervision produces predictions close to a single connected artery tree, a desirable property for clinical interpretation and downstream geometric analysis. The feed-forward pipeline enables sub-second reconstruction without per-case optimisation.

Figure 5(B) provides an initial qualitative application to real X-ray coronary angiography, but it does not constitute systematic clinical validation. Sequentially acquired clinical monoplane views can retain cardiac deformation even after phase matching. Future work should evaluate larger clinical cohorts and address inter-view misalignment caused by motion and gantry displacement (Tu et al., 2010) through motion-aware reconstruction and geometry calibration. A central open question is how to evaluate 3D reconstruction accuracy without a 3D artery label; paired acquisitions of CT and X-ray coronary angiography or expert assessment may therefore be required.

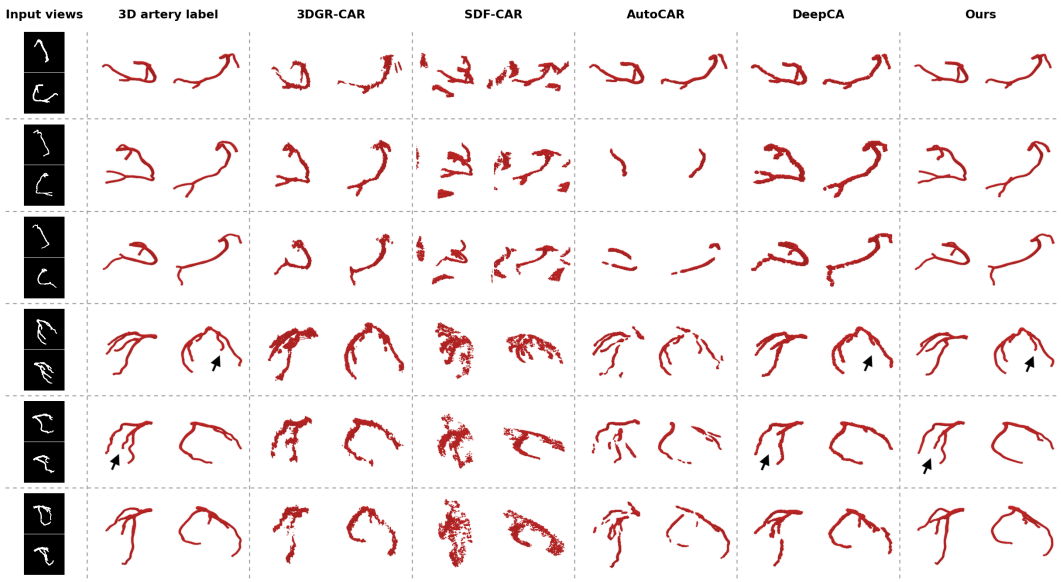


Figure 4: Qualitative two-view reconstruction. Columns show the input masks, 3D label, and the method outputs; rows show three RCA cases and three LCA cases, each rendered from two viewpoints. Arrows mark branches poorly visible in the input projections.

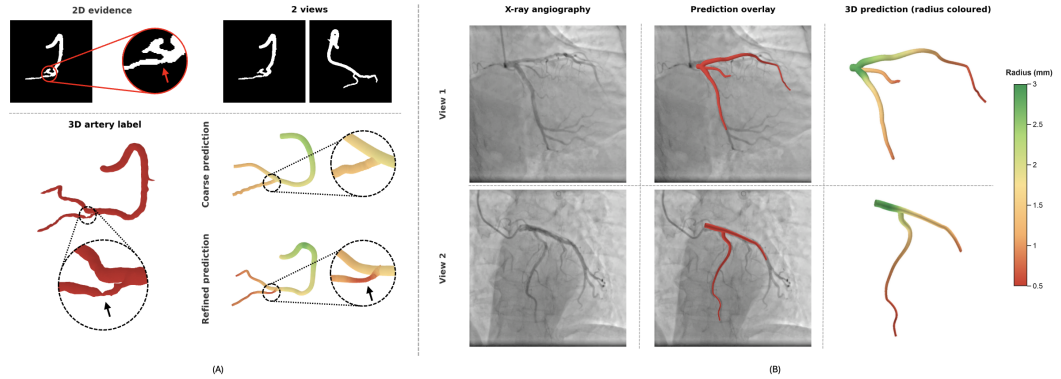


Figure 5: Qualitative radius reconstruction. (A) A diseased ASOCA (Gharleghi et al., 2023) case with a visually identified region of local artery narrowing. The 3D label, simulated masks, and radius-coloured coarse and refined prediction are shown. (B) A real two-view X-ray angiography case from the AutoCAR dataset (Zhu et al., 2025). The middle column overlays the prediction on each angiographic view, and the final column shows the radius-coloured 3D prediction.

The learned coronary anatomy patterns can support a plausible reconstruction when projection evidence is incomplete, but they cannot reliably identify a specific stenosis that is concealed by overlap or foreshortening in every available view. Moreover, without clinical stenosis annotations, our current examples assess radius reconstruction rather than diagnostic stenosis detection. Further studies using expert lesion annotations and physiological measurements such as fractional flow reserve are therefore required to determine whether reconstructed radius changes correspond to clinically significant stenoses.

Finally, the present model requires known 3D artery labels for supervised training. Developing unsupervised or self-supervised objectives from real angiographic views could reduce this dependence; the current model may also provide initial pseudo-3D labels for datasets that contain only 2D X-ray coronary angiography images.

AI USE STATEMENT

In this work, generative AI tools were used to help design or provide feedback on research methodology or experiments and implement methods. They were also used to create or modify scientific figures or images, suggest experimental parameters, create or edit software code, support the creation of artefacts, summarise or analyse existing literature, discover research topics or identify gaps, support brainstorming and sourcing/searching for information, edit a research paper to improve readability, identify relevant literature, format references, and propose a title or keywords for a research paper. The authors reviewed all AI-assisted work, checked literature-related outputs against the original sources, and inspected and tested AI-assisted code. All experimental decisions, analyses, reported results, and final manuscript content were verified by the authors, who take full responsibility for this work.

REFERENCES

Xueming Fu, Yingtai Li, Fenghe Tang, Jun Li, Mingyue Zhao, Gao-Jun Teng, and S. Kevin Zhou. 3DGR-CAR: Coronary artery reconstruction from ultra-sparse 2d x-ray views with a 3d gaussians representation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, volume 15007 of *Lecture Notes in Computer Science*, pp. 14–24. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72104-5_2.

Ramtin Gharleghi, Dona Adikari, Katy Ellenberger, Michael Webster, Chris Ellis, Arcot Sowmya, Sze-Yuan Ooi, and Susann Beier. Annotated computed tomography coronary angiogram images and associated data of normal and diseased arteries. *Scientific Data*, 10(1):128, 2023. doi: 10.1038/s41597-023-02016-2.

Zhanqiang Guo, Zimeng Tan, Jianjiang Feng, and Jie Zhou. VesselDiffusion: 3d vascular structure generation based on diffusion model. *IEEE Transactions on Medical Imaging*, 44(9):3845–3857, 2025. doi: 10.1109/TMI.2025.3568602.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Kritika Iyer, Brahmajee K. Nallamothu, C. Alberto Figueroa, and Raj R. Nadakuditi. A multi-stage neural network approach for coronary 3d reconstruction from uncalibrated x-ray angiography images. *Scientific Reports*, 13:17603, 2023. doi: 10.1038/s41598-023-44633-2.

A. K. Klein, F. Lee, and A. A. Amini. Quantitative coronary angiography with deformable spline models. *IEEE Transactions on Medical Imaging*, 16(5):468–482, 1997. doi: 10.1109/42.640737.

J. L. Klein, J. G. Hoff, J. W. Peifer, R. Folks, C. D. Cooke, S. B. King, and E. V. Garcia. A quantitative evaluation of the three dimensional reconstruction of patients' coronary arteries. *International Journal of Cardiac Imaging*, 14(2):75–87, 1998. doi: 10.1023/A:1005903705300.

Kirsten W. H. Maas, Danny Ruijters, Anna Vilanova, and Nicola Pezzotti. NeRF-CA: Dynamic reconstruction of x-ray coronary angiography with extremely sparse-views. *IEEE Transactions on Visualization and Computer Graphics*, 31(10):8782–8795, 2025. doi: 10.1109/TVCG.2025.3579162.

Kirsten W. H. Maas, Danny Ruijters, Nicola Pezzotti, and Anna Vilanova. NerT-CA: Efficient dynamic reconstruction from sparse-view x-ray coronary angiography. In *Reconstruction, Imaging, and Mixed Reality in Healthcare / Graphs in Biomedical Image Analysis*, volume 16150 of *Lecture Notes in Computer Science*, pp. 13–22, 2026. doi: 10.1007/978-3-032-06103-4_2.

Ahmed Reda, Mohamed Ashraf, Mohamed G. Abdelkader, Yasser Abuouf, and Muhamed Albadawi. SDF-CAR: 3d coronary artery reconstruction from two views with a hybrid SDF-occupancy implicit representation. Full paper under review for MIDL 2026, 2026. URL https://openreview.net/pdf?id=bfjv51bxKJ.

Guy Shechter, Frédéric Devernay, Eve Coste-Manière, Arshed Quyyumi, and Elliot R. McVeigh. Three-dimensional motion tracking of coronary arteries in biplane cineangiograms. *IEEE Transactions on Medical Imaging*, 22(4):493–503, 2003. doi: 10.1109/TMI.2003.809090.

Shengxian Tu, Gerhard Koning, Wouter Jukema, and Johan H. C. Reiber. Assessment of obstruction length and optimal viewing angle from biplane x-ray angiograms. *The International Journal of Cardiovascular Imaging*, 26(1):5–17, 2010. doi: 10.1007/s10554-009-9509-3.

Arso M. Vukicevic, Serkan Cimen, Nikola Jagic, Goran Jovicic, Alejandro F. Frangi, and Nenad Filipovic. Three-dimensional reconstruction and NURBS-based structured meshing of coronary arteries from conventional x-ray angiography projection images. *Scientific Reports*, 8(1):1711, 2018. doi: 10.1038/s41598-018-19440-9.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.html.

Yiying Wang, Abhirup Banerjee, Robin P. Choudhury, and Vicente Grau. DeepCA: Deep learning-based 3d coronary artery tree reconstruction from two 2d non-simultaneous x-ray angiography projections. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 337–346, 2025b. URL https://openaccess.thecvf.com/content/WACV2025/html/Wang_DeepCA_Deep_Learning-Based_3D_Coronary_Artery_Tree_Reconstruction_from_Two_WACV_2025_paper.html.

An Zeng, Chunbiao Wu, Guisen Lin, Wen Xie, Jin Hong, Meiping Huang, Jian Zhuang, Shanshan Bi, Dan Pan, Najeeb Ullah, Kaleem Nawaz Khan, Tianchen Wang, Yiyu Shi, Xiaomeng Li, and Xiaowei Xu. ImageCAS: A large-scale dataset and benchmark for coronary artery segmentation based on computed tomography angiography images. *Computerized Medical Imaging and Graphics*, 109:102287, 2023. doi: 10.1016/j.compmedimag.2023.102287.

Yinheng Zhu, Yong Wang, Chunxia Di, Hanghang Liu, Fangzhou Liao, and Shaohua Ma. Sparse and transferable three-dimensional dynamic vascular reconstruction for instantaneous diagnosis. *Nature Machine Intelligence*, 7:730–742, 2025. doi: 10.1038/s42256-025-01025-7.

A Appendix

A.1 Dataset and Target-Construction Details

The ImageCAS coronary annotations (Zeng et al., 2023) are provided as NIfTI volumes. We use the x-, y-, and z-axis spacing metadata to express all centreline coordinates, radii, and distances in millimetres. Starting from the 1,000 source cases, each labelled artery volume is automatically separated into RCA and LCA components and every separated component is inspected manually. A component is excluded if its RCA/LCA identity cannot be determined reliably, if the two artery systems cannot be separated, or if annotation defects prevent construction of a valid target. The resulting RCA and LCA collections are component cohorts rather than disjoint patient cohorts: retained RCA and LCA components may originate from the same ImageCAS case, so their counts do not represent completely different patients. These exclusions may bias the retained cohorts towards cases with clearer artery annotations and should be considered when interpreting generalisation to the complete clinical population.

Retained components are skeletonised in physical coordinates and converted into an initial centreline graph using local node connections and a minimum spanning tree, following the broad graph-construction procedure used by AutoCAR (Zhu et al., 2025). Because automatic conversion can fail for tortuous vessels, rapidly changing radii, and annotation defects, manual supervision removes extra branches, repairs erroneous connections, and verifies branch identity and ordering. Difficult geometry is retained whenever a reliable target can be produced after correction. For each side branch, the first centreline point is set to its attachment point on the corresponding main branch, explicitly encoding branch connectivity in the target vessel code. Radius is obtained from the distance transform of the CT artery mask, with each cross-section approximated by a circular lumen profile. Every branch is resampled from its proximal root or attachment to its distal endpoint to produce $N = 200$ ordered centreline-and-radius samples and is fitted with $K_{\mathrm{cp}} = 20$ B-spline control points. This quality-control process produces 750 RCA and 750 LCA cases, split at patient level into 600 training, 75 validation, and 75 test cases.

For the qualitative narrowing example in Figure 5(A), we additionally use one diseased case from the ASOCA dataset (Gharleghi et al., 2023). ASOCA provides 40 expert-annotated CT coronary angiography volumes, comprising 20 healthy participants and 20 participants with confirmed coronary artery disease. This ASOCA case is used only for qualitative evaluation and is not included in model training or the quantitative test cohorts.

For RCA targets, the RCA trunk is the first branch and retained side branches are sorted by the normalised arc-length position of their attachment point while traversing the RCA trunk from proximal to distal. For LCA targets, the LM–LAD path is first, the LCX path is second, and the remaining side branches are sorted first along LAD and then along LCX, again by proximal-to-distal attachment position. Every retained ImageCAS LCA case contains a distinct LM artery. The ordering gives each optional side branch a deterministic target entry without assigning it a fixed anatomical name. For RCA, we retained a maximum of 7 branches; for LCA, we retained a maximum of 13 branches. For each case of RCA or LCA, unused side-branch entries are zero-padded and assigned an existence label of zero.

Seven clinically motivated projection anchors are defined separately for RCA and LCA, as listed in Table 3. Here, RAO and LAO denote the primary right and left anterior oblique rotations, while CRA and CAU denote the secondary cranial and caudal angulations; AP denotes zero primary rotation. These values are anchor directions rather than the only admissible acquisition angles, so nearby angular settings belong to the same clinical view family. For both RCA and LCA, the primary and secondary angles are independently varied within $\pm 10^{\circ}$ of each artery-specific anchor.

Patient coordinates and projection centre. Vessel coordinates use the source-volume left–anterior–superior (LAS) patient frame: the positive x, y, and z axes point towards the patient's left, anterior, and superior directions, respectively. No additional standing–supine rotation is applied. For each case, the projection isocentre is placed at the centroid of the reference artery surface. The reference is the main RCA surface for RCA and the combined LM–LAD and LCX surfaces for LCA. If this centroid is $\boldsymbol{o}$ in millimetres, the camera receives the centred point $\bar{\boldsymbol{x}} = 10^{-3}(\boldsymbol{x} - \boldsymbol{o})$ in metres, so the isocentre is the world origin $\boldsymbol{O} = (0, 0, 0)^{\top}$.

Table 3: Clinical acquisition-direction anchors used to generate the seven candidate projections for each artery. Each entry reports the primary RAO/LAO rotation followed by the secondary CRA/CAU angulation.

View	RCA anchor	LCA anchor
1	LAO 40°, CRA 10°	RAO 25°, CAU 35°
2	RAO 75°, CRA 10°	LAO 5°, CAU 30°
3	AP, CRA 25°	AP, CAU 10°
4	RAO 20°, CRA 20°	RAO 40°, CRA 25°
5	LAO 20°	LAO 20°, CRA 30°
6	RAO 75°, CAU 20°	LAO 40°, CRA 20°
7	LAO 60°	LAO 5°, CRA 40°

Clinical angles and camera rotation. We use a positive clinical primary angle for LAO and a negative angle for RAO; a positive secondary angle denotes CRA and a negative angle denotes CAU. AP has zero primary angle. The preprocessing converts these clinical angles (a_k, b_k) to the stored projector angles (θ_k, ϕ_k). The conversion is $\theta_k = \mathrm{wrap}(90^\circ - a_k)$ and $\phi_k = 90^\circ - b_k$, where wrap maps an angle to $[-180^\circ, 180^\circ)$.

Source, detector, and projection. The cone-beam projector uses a 256×256-pixel flat detector, a source-to-detector distance of $0.90\,\mathrm{m}$, and a source-to-isocentre distance of $0.75\,\mathrm{m}$. The detector pixel spacing is $0.55\,\mathrm{mm/pixel}$ for RCA and $0.65\,\mathrm{mm/pixel}$ for LCA.

A.2 Model Details, Training, and Evaluation Protocols

This section collects the architectural, target-fitting, training, and test-time optimisation details that complement the main Methods section.

A.2.1 B-spline Representation and Target Fitting

For each artery branch m, the ordered dense centreline samples $\{\boldsymbol{c}_{m,n}\}_{n=1}^{N}$ are parameterised by a cubic B-spline with $K_{\mathrm{cp}} = 20$ controls. The fixed basis matrix $\boldsymbol{A} \in \mathbb{R}^{N \times K_{\mathrm{cp}}}$ is evaluated at $t_n = (n-1)/(N-1)$ and has entries $A_{n,j} = B_{j,3}(t_n)$. The basis uses the open-uniform clamped knot vector $\boldsymbol{u} = (0, 0, 0, 0, 1/17, \ldots, 16/17, 1, 1, 1, 1)$. The branch control matrix is fitted offline by endpoint-constrained least squares:

$$\begin{aligned} \boldsymbol{P}_m^\star &= \arg\min_{\boldsymbol{P}_m} \|\boldsymbol{C}_m - \boldsymbol{A}\boldsymbol{P}_m\|_F^2 , \\ \text{subject to} \quad \boldsymbol{A}_{1,:}\boldsymbol{P}_m &= \boldsymbol{c}_{m,1}, \\ \boldsymbol{A}_{N,:}\boldsymbol{P}_m &= \boldsymbol{c}_{m,N}, \end{aligned} \tag{9}$$

where $\boldsymbol{P}_m \in \mathbb{R}^{K_{\mathrm{cp}} \times 3}$ is the control-point matrix, $\boldsymbol{C}_m \in \mathbb{R}^{N \times 3}$ contains the dense target coordinates, and $\|\cdot\|_F$ is the Frobenius norm. The decoded approximation is $\widetilde{\boldsymbol{C}}_m = \boldsymbol{A}\boldsymbol{P}_m^\star$.

Before training the image-to-artery model, we test whether the compact spline can represent the accepted ground-truth vessels. For each candidate control count, target controls are fitted and decoded to $N = 200$ centreline points. Fitting error is the mean paired Euclidean distance between decoded and original points at corresponding normalised-arc-length positions. Vessel clDice is computed using the unchanged dense pointwise radii, thereby isolating centreline-parameterisation fidelity. Table 4 quantifies the approximation–compactness trade-off. With $K_{\mathrm{cp}} = 20$, the overall fitting error is 0.164 ± 0.108 mm and vessel clDice is $99.92 \pm 0.62\%$, using 60 centreline parameters per branch instead of 600 dense coordinates. Table 5 complements this offline analysis by evaluating whether the B-spline parameterisation also benefits learned two-view coarse prediction and projection-guided refinement relative to directly predicting and refining all $N = 200$ dense centreline points.

Table 4: Offline B-spline representation fidelity. Entries are mean ± sample standard deviation. Centreline fitting error is reported in millimetres and vessel clDice as a percentage. The RCA and LCA statistics each use 750 cases. The overall statistic pools 1,500 source-case means, with each LCA case contributing the arithmetic mean of its LM–LAD and LCX branch metrics.

K_{cp}	Centreline fitting error (mm) ↓				Vessel clDice (%) ↑			
	RCA	LM–LAD	LCX	Overall	RCA	LM–LAD	LCX	Overall
8	1.18 (±0.72)	0.65 (±0.40)	0.81 (±0.38)	0.95 (±0.60)	86.21 (±17.91)	96.98 (±7.11)	92.78 (±9.66)	90.55 (±14.21)
12	0.56 (±0.41)	0.31 (±0.21)	0.46 (±0.26)	0.47 (±0.33)	97.35 (±6.61)	99.47 (±2.24)	98.01 (±4.45)	98.04 (±5.08)
20	0.19 (±0.13)	0.11 (±0.08)	0.17 (±0.11)	0.16 (±0.11)	99.88 (±0.84)	100.00 (±0.06)	99.94 (±0.49)	99.92 (±0.62)
Dense coordinates	0.00	0.00	0.00	0.00	100.00	100.00	100.00	100.00

Table 5: Effect of centreline representation on coarse and refined reconstruction using two input views. The dense-centreline variant directly predicts and refines the $N = 200$ ordered centreline points, whereas the B-spline variant predicts and refines $K_{\mathrm{cp}} = 20$ control points before decoding them to the same number of dense points. All other model and evaluation settings are unchanged. CD denotes Chamfer distance, and CC count denotes connected-component count. Values are mean ± standard error; **bold** indicates the better representation within each matched prediction stage.

Stage	Centreline representation	RCA				LCA			
		Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓	Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓
Coarse	Dense centreline	10.22 (±0.74)	10.23 (±0.98)	13.81 (±0.46)	**1.284** (±0.056)	8.97 (±0.53)	8.29 (±0.76)	16.09 (±0.59)	2.276 (±0.125)
	B-spline	**19.47** (±1.12)	**20.40** (±1.56)	**9.70** (±0.41)	1.324 (±0.064)	**14.92** (±0.77)	**13.94** (±0.90)	**12.37** (±0.51)	**1.605** (±0.097)
Refined	Dense centreline	22.18 (±1.07)	26.87 (±1.57)	10.36 (±0.47)	1.189 (±0.050)	15.21 (±0.68)	15.77 (±0.91)	13.87 (±0.58)	1.526 (±0.087)
	B-spline	**66.92** (±1.50)	**80.62** (±1.92)	**3.16** (±0.27)	**1.054** (±0.026)	**49.45** (±1.32)	**61.30** (±1.93)	**5.98** (±0.47)	**1.053** (±0.026)

A.2.2 Coarse Model Encoder Details

This section provides the implementation details of the frozen VGGT encoder used by the coarse predictor in Subsection 3.3.

VGGT backbone. The VGGT backbone (Wang et al., 2025a) processes each input view independently and produces visual features

$$\boldsymbol{F}_k = f_{\mathrm{VGGT}}(\boldsymbol{I}_k), \qquad \boldsymbol{F}_k \in \mathbb{R}^{T_{\mathrm{VGGT}} \times d_{\mathrm{VGGT}}}. \tag{10}$$

A learned projection and layer normalisation map these features to the common dimension:

$$\boldsymbol{H}_k^I = \mathrm{LN}\left(\boldsymbol{F}_k \boldsymbol{W}_{\mathrm{VGGT}} + \boldsymbol{\beta}_{\mathrm{VGGT}}\right) \in \mathbb{R}^{T_{\mathrm{VGGT}} \times d}, \tag{11}$$

where $\boldsymbol{W}_{\mathrm{VGGT}} \in \mathbb{R}^{d_{\mathrm{VGGT}} \times d}$ and $\boldsymbol{\beta}_{\mathrm{VGGT}} \in \mathbb{R}^d$ are learned. Under the shared notation of the main paper, $T_{\mathrm{img}} = T_{\mathrm{VGGT}}$.

To assess the effect of the image backbone, Table 6 compares the frozen VGGT backbone (Wang et al., 2025a) with ResNet-101 (He et al., 2016) using two input views. The ResNet alternative uses ResNet-101 without fine-tuning its backbone parameters.

A.2.3 Branch Decoder Details

The parallel coarse decoder uses M trainable branch queries $\boldsymbol{q}_m \in \mathbb{R}^d$ and the query matrix $\boldsymbol{Q}_{\mathrm{query}} = [\boldsymbol{q}_1; \ldots; \boldsymbol{q}_M]$. Every query attends to the multiview memory but not to another branch query. Using $\boldsymbol{E}_{\mathrm{view}}$ as the cross-attention keys and values, the shared decoder produces branch tokens

$$[\boldsymbol{b}_1; \ldots; \boldsymbol{b}_M] = f_{\text{cross-attn}}(\boldsymbol{Q}_{\mathrm{query}}, \boldsymbol{E}_{\mathrm{view}}), \qquad \boldsymbol{b}_m \in \mathbb{R}^d. \tag{12}$$

The branch queries are decoded independently and in parallel and correspond to the ordered target entries defined in Subsection 3.2. Query m therefore retrieves evidence for branch entry m, but optional entries do not denote fixed, named anatomical side-branch categories. An existence prediction

Table 6: Two-view backbone comparison for RCA and LCA coarse and refined reconstruction. CD denotes Chamfer distance, and CC count denotes connected-component count. Values are mean ± standard error.

Backbone	Output	RCA				LCA			
		Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓	Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓
ResNet-101	Coarse	18.94 (±1.15)	19.94 (±1.43)	9.37 (±0.37)	1.428 (±0.097)	13.88 (±1.78)	13.69 (±0.89)	12.99 (±0.32)	1.878 (±0.097)
ResNet-101	Refined	60.62 (±1.29)	72.80 (±1.61)	4.06 (±0.47)	1.087 (±0.029)	45.19 (±1.71)	56.33 (±1.50)	6.71 (±0.44)	1.102 (±0.064)
VGGT	Coarse	19.47 (±1.12)	20.40 (±1.56)	9.70 (±0.41)	1.324 (±0.064)	14.92 (±0.77)	13.94 (±0.90)	12.37 (±0.51)	1.605 (±0.097)
VGGT	Refined	66.92 (±1.50)	80.62 (±1.92)	3.16 (±0.27)	1.054 (±0.026)	49.45 (±1.32)	61.30 (±1.93)	5.98 (±0.47)	1.053 (±0.026)

determines which optional branch outputs are active for an individual reconstruction. The prediction of side branch m is directly supervised by target entry m; no Hungarian matching is used.

The control head is

$$g_{\mathrm{cp}} : \mathbb{R}^d \to \mathbb{R}^{3K_{\mathrm{cp}}}, \qquad \hat{\boldsymbol{P}}_m^{(0)} = \mathrm{reshape}(g_{\mathrm{cp}}(\boldsymbol{b}_m)). \tag{13}$$

The raw radius head predicts N log-radii,

$$g_{\mathrm{rad}} : \mathbb{R}^d \to \mathbb{R}^N, \qquad \hat{\boldsymbol{r}}_m^{(0)} = \exp(g_{\mathrm{rad}}(\boldsymbol{b}_m)), \tag{14}$$

with the log values clamped to a finite numerical range before exponentiation. The branch-existence head produces one logit per query,

$$g_{\mathrm{exist}} : \mathbb{R}^d \to \mathbb{R}, \qquad z_m = g_{\mathrm{exist}}(\boldsymbol{b}_m). \tag{15}$$

At inference, the main-branch entry is forced active and side-branch probabilities are $\hat{e}_m = \sigma(z_m)$. The prediction heads are shared across all branch queries, and every output is decoded in the same world frame. Thus, M is a fixed maximum number of query outputs, while the predicted active branch count can vary between cases. We use maximum query counts of $M_{\mathrm{RCA}} = 7$ and $M_{\mathrm{LCA}} = 13$. The required primary branches are always active. An optional branch is retained at inference when $\sigma(z_m) \geq 0.5$.

A.2.4 Projection-Guided Geometry and Radius Refiners

B-spline geometry refinement. The coarse predictor supplies a population-level artery estimate, but ambiguity in the sparse input views can leave case-specific centreline discrepancies. The geometry refiner corrects these discrepancies directly in the compact B-spline control-point space while holding the coarse radii $\hat{\boldsymbol{r}}_m^{(0)}$ fixed. Its inputs are the coarse controls $\hat{\boldsymbol{P}}^{(0)}$, branch tokens, coronary mask images, acquisition directions, spatial image features extracted by the frozen backbone, and the known differentiable projection $\Pi_k : \mathbb{R}^3 \to \mathbb{R}^2$ for each view k. The same frozen-backbone features used by the coarse predictor are therefore reused during geometry refinement; the backbone itself remains frozen.

Because a B-spline control point does not generally lie on the decoded curve, each control index j is associated with an on-curve anchor. The anchor index is the sampled centreline position at which the corresponding basis function is strongest:

$$n_j = \arg\max_n A_{n,j}. \tag{16}$$

At refinement stage s, the current decoded anchor is

$$\boldsymbol{a}_{m,j}^{(s)} = \sum_{\ell=1}^{K_{\mathrm{cp}}} A_{n_j,\ell} \hat{\boldsymbol{p}}_{m,\ell}^{(s)}. \tag{17}$$

Here, $s \in \{0, \ldots, S_g - 1\}$ denotes the geometry-refinement stage. The anchor is projected to view k as $\boldsymbol{u}_{m,j,k}^{(s)} = \Pi_k(\boldsymbol{a}_{m,j}^{(s)})$. Around $\boldsymbol{u}_{m,j,k}^{(s)}$, the implementation samples an odd-sized input-mask

patch, local learned image features, the mask distance transform, and its two-dimensional gradient. It also extracts the backbone feature at the corresponding image region: from the spatial VGGT patch-token map in the final model, or from the ResNet feature map in the backbone comparison. Collectively, these samples form the local multiview projection evidence for control point (m, j). Samples outside the detector or belonging to padded views are marked invalid.

The evidence encoder maps the local samples to dimension d. A refinement query combines the current control position, anchor position, branch token, control-index embedding, branch-index embedding, and recurrent state. Multi-head attention aggregates the encoded samples over valid views. The resulting state is passed to a residual head:

$$\Delta\hat{\boldsymbol{p}}_{m,j}^{(s)} = \alpha_g \tanh\left(g_g(\boldsymbol{z}_{m,j}^{(s)})\right) \in \mathbb{R}^3, \tag{18}$$

where $\boldsymbol{z}_{m,j}^{(s)} \in \mathbb{R}^d$ is the evidence-conditioned state, $g_g : \mathbb{R}^d \to \mathbb{R}^3$ is the learned residual head, and α_g is the maximum update magnitude in millimetres. The controls are updated by

$$\hat{\boldsymbol{P}}^{(s+1)} = \hat{\boldsymbol{P}}^{(s)} + \Delta\hat{\boldsymbol{P}}^{(s)}. \tag{19}$$

The output layer parameters are shared across the configured S_g stages. After each stage, the updated B-spline is decoded, its anchors are reprojected, and new image evidence is sampled for the next stage. The radius vector remains fixed throughout this module. Geometry refinement is therefore a fixed-depth learned computation at inference rather than per-case gradient-based optimisation.

The reported geometry-refiner configuration uses $S_g = 4$ stages, 17×17 local patches, a 256-dimensional evidence state, and a maximum per-stage control update of $\alpha_g = 7\,\mathrm{mm}$.

Dense-radius refinement. The radius refiner is enabled only with B-spline centrelines and raw dense-radius prediction. It receives the geometry-refined vessel, holds the refined centreline $\hat{\boldsymbol{C}}_m^{(S_g)}$ fixed, and updates only the dense radii. At radius-refinement stage $u \in \{0, \ldots, S_r - 1\}$, the current centreline-and-radius model is differentiably rendered into every valid input view.

For projected centreline point (m, n), finite differences along the projected curve estimate a unit tangent. Rotating this tangent by 90° gives the transverse image-plane normal. Let L be the odd number of profile samples and w_{prof} the profile half-width in pixels. The sampling grid is

$$\boldsymbol{u}_{m,n,k,\ell} = \Pi_k(\hat{\boldsymbol{c}}_{m,n}) + \delta_\ell \boldsymbol{n}_{m,n,k}, \qquad \delta_\ell \in [-w_{\mathrm{prof}}, w_{\mathrm{prof}}], \quad \ell \in \{1, \ldots, L\}, \tag{20}$$

where $\boldsymbol{n}_{m,n,k}$ is the transverse unit normal. Four profiles are sampled: the observed input mask, the rendered vessel mask, their signed difference, and their absolute difference. These profiles expose whether the rendered vessel is too narrow or too wide around the current projected centreline while reducing interference from distant parts of the projected coronary tree. They are concatenated with the encoded view information and mapped to the model dimension.

For point (m, n), the query is formed from the branch token, branch-index and centreline-point-index embeddings, encoded current radius, and recurrent point state. Multi-head attention aggregates the valid per-view profiles. Two one-dimensional convolutional blocks then propagate context along the ordered centreline. A zero-initialised scalar residual head predicts

$$\Delta\hat{r}_{m,n}^{(u)} = \alpha_r \tanh\left(g_r(\boldsymbol{z}_{m,n}^{(u)})\right), \tag{21}$$

where u is the radius-refinement stage, $\boldsymbol{z}_{m,n}^{(u)} \in \mathbb{R}^d$ is the evidence-conditioned point state, $g_r : \mathbb{R}^d \to \mathbb{R}$ is the residual head, and α_r bounds one correction in millimetres. The update is

$$\hat{r}_{m,n}^{(u+1)} = \max\left(r_{\min}, \hat{r}_{m,n}^{(u)} + \Delta\hat{r}_{m,n}^{(u)}\right). \tag{22}$$

The updated surface is rerendered and new transverse evidence is sampled before the next of the S_r stages. Because centreline coordinates are detached from this pathway, radius refinement cannot change vessel trajectory to reduce its loss. After the configured geometry and radius stages, the final refined vessel code is

$$\hat{\boldsymbol{V}}_{\mathrm{refined}} = \left\{(\hat{\boldsymbol{c}}_{m,n}^{(S_g)}, \hat{r}_{m,n}^{(S_r)})\right\}_{m,n}. \tag{23}$$

The reported radius-refiner configuration uses $S_r = 3$ stages, $L = 25$ transverse samples spanning $w_{\mathrm{prof}} = 16$ pixels on either side of the projected centreline, a maximum update of $\alpha_r = 1\,\mathrm{mm}$ per stage, and minimum radius $r_{\min} = 0.05\,\mathrm{mm}$.

Table 7: Effect of branch-subset augmentation on two-view coarse prediction and geometry refinement. CD denotes Chamfer distance, and CC count denotes connected-component count. Values are mean ± standard error.

Prediction stage	Branch-subset augmentation	RCA				LCA			
		Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓	Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓
Coarse prediction	×	11.38 (±0.95)	10.89 (±1.09)	13.42 (±0.52)	1.338 (±0.070)	9.82 (±0.54)	8.81 (±0.59)	14.70 (±0.49)	1.882 (±0.082)
Coarse prediction	✓	19.47 (±1.12)	20.40 (±1.56)	9.70 (±0.41)	1.324 (±0.064)	14.92 (±0.77)	13.94 (±0.90)	12.37 (±0.51)	1.605 (±0.097)
Geometry refiner	×	46.57 (±1.45)	58.45 (±2.04)	6.13 (±0.43)	1.068 (±0.029)	37.52 (±1.04)	44.57 (±1.51)	8.01 (±0.45)	1.211 (±0.054)
Geometry refiner	✓	65.89 (±1.44)	80.84 (±1.90)	3.16 (±0.27)	1.068 (±0.029)	48.43 (±1.29)	61.38 (±1.93)	5.98 (±0.47)	1.053 (±0.026)

A.2.5 Within-Case Branch-Subset Training

Branch-count variation in the original dataset is entangled with differences in case-specific anatomy. To provide direct supervision for optional branch presence, we construct multiple branch-subset variants from each training case while holding its underlying anatomy fixed.

For a case with B ordered branches, every variant retains the required main path or paths: branch 1 for RCA and branches 1–2 for LCA. Starting from the complete artery tree, optional side branches are successively removed in reverse canonical order until only the required main path or paths remain. An RCA case therefore produces B variants, while an LCA case produces $B - 1$ variants, including the complete-tree and main-path-only versions. For every variant, we regenerate the projected masks and branch-existence labels while preserving the geometry targets and ordered identities of retained branches.

All variants from the same case are collated in one optimisation batch and use the same selected input views. The training loader visits every physical case once per epoch, and every corresponding branch-subset variant is included; no Bernoulli sampling probability is used. The variants are presented together rather than introduced sequentially as a curriculum. This protocol is applied when training both the coarse predictor and the geometry refiner.

Table 7 shows that within-case branch-subset training improves Dice, clDice, and Chamfer distance for both prediction stages and artery cohorts. For branch-existence prediction, it increases accuracy from 92.95% to 94.63% and F1 from 88.37% to 92.48%. The improvement is driven primarily by higher precision and specificity, indicating fewer false-positive side-branch predictions, although recall decreases modestly. These results suggest that varying branch presence while preserving patient anatomy helps the model learn that side branches are optional.

A.2.6 Direct Per-Scene B-spline Optimisation Baseline

The direct-optimisation baseline starts from the same coarse control points $\hat{\boldsymbol{P}}^{(0)}$ as the learned geometry refiner. For each test case, gradient-based optimisation updates only the B-spline controls to reduce the selected 2D projection objective against the input masks:

$$\boldsymbol{P}^{\star}_{\mathrm{opt}} = \arg\min_{\boldsymbol{P}} \mathcal{L}_{\mathrm{proj}} \left(\mathrm{Render}(\boldsymbol{A}\boldsymbol{P}, \hat{\boldsymbol{r}}^{(0)}), \{\boldsymbol{I}_k\}_{k=1}^{K} \right). \tag{24}$$

The coarse radius is fixed. The number of optimisation iterations and all hyperparameters are determined during method development and fixed before test-set evaluation. This baseline isolates whether the learned geometry refiner provides a better accuracy–runtime trade-off than test-time optimisation from the same initialisation.

Table 10 compares direct optimisation with the learned geometry and radius refiners from a common two-view coarse initialisation for RCA and LCA, including their post-coarse processing time.

A.3 Implementation Details and Ablation Studies

A.3.1 Experiment Details

The models are implemented in PyTorch and trained with one GPU per run. We use AdamW with an initial learning rate of 10^{-4}, weight decay of 10^{-4}, cosine learning-rate decay to 10^{-6}, and a gradient-norm clipping threshold of 1.0. The coarse predictor, geometry refiner, and radius refiner all use a batch size of 16. The coarse predictor and geometry refiner are trained for at most 200 epochs, and the radius refiner for at most 100 epochs. The frozen image backbone is initialised from the pretrained `facebook/VGGT-1B` checkpoint, and its features are precomputed before predictor training. The test set is evaluated only after model and hyperparameter selection. All reported runtime measurements use a single NVIDIA GeForce RTX 4090 GPU.

A.3.2 Evaluation Implementation Details

All volumetric metrics are computed on the common isotropic $0.5\,\mathrm{mm}$ evaluation grid defined in Subsection 3.6. The ground-truth mask is resampled by nearest-neighbour interpolation, while the predicted centreline and radii are rasterised as the union of filled, radius-varying tubular segments on this grid. The resulting voxelised prediction is used consistently for Dice, clDice, and connected-component counting.

Let $\widehat{\mathcal{V}}$ and $\mathcal{V}$ denote the foreground-voxel sets of the predicted and ground-truth arteries, respectively. Their volumetric overlap is measured by

$$\mathrm{Dice} = \frac{2|\widehat{\mathcal{V}} \cap \mathcal{V}|}{|\widehat{\mathcal{V}}| + |\mathcal{V}|}. \tag{25}$$

For clDice, let $\mathcal{S}(\cdot)$ denote the one-voxel 3D centreline obtained by skeletonisation. Topology precision, topology sensitivity, and clDice are

$$\begin{aligned} T_{\mathrm{prec}} &= \frac{|\mathcal{S}(\widehat{\mathcal{V}}) \cap \mathcal{V}|}{|\mathcal{S}(\widehat{\mathcal{V}})|}, \\ T_{\mathrm{sens}} &= \frac{|\mathcal{S}(\mathcal{V}) \cap \widehat{\mathcal{V}}|}{|\mathcal{S}(\mathcal{V})|}, \\ \mathrm{clDice} &= \frac{2T_{\mathrm{prec}}T_{\mathrm{sens}}}{T_{\mathrm{prec}} + T_{\mathrm{sens}}}. \end{aligned} \tag{26}$$

Each ground-truth RCA or LCA tree contains exactly one connected component because all retained side branches attach to the corresponding main branch. Connected-component count evaluates whether a prediction preserves this expected structure. It is computed from the same $0.5\,\mathrm{mm}$ voxelised prediction used for Dice and clDice. We apply 3D connected-component labelling to the foreground set $\widehat{\mathcal{V}}$ and report

$$N_{\mathrm{CC}} = \left|\mathrm{CC}(\widehat{\mathcal{V}})\right|, \tag{27}$$

where $\mathrm{CC}(\widehat{\mathcal{V}})$ is the set of connected foreground components. A value of one indicates a single connected prediction, while larger values indicate fragmentation.

Finally, let $\widehat{\mathcal{C}}$ and $\mathcal{C}$ be the predicted and ground-truth centreline point sets in millimetre-valued world coordinates. We use the symmetric Chamfer distance

$$\mathrm{CD}(\widehat{\mathcal{C}}, \mathcal{C}) = \frac{1}{|\widehat{\mathcal{C}}|} \sum_{\hat{\boldsymbol{c}} \in \widehat{\mathcal{C}}} \min_{\boldsymbol{c} \in \mathcal{C}} \|\hat{\boldsymbol{c}} - \boldsymbol{c}\|_2 + \frac{1}{|\mathcal{C}|} \sum_{\boldsymbol{c} \in \mathcal{C}} \min_{\hat{\boldsymbol{c}} \in \widehat{\mathcal{C}}} \|\boldsymbol{c} - \hat{\boldsymbol{c}}\|_2. \tag{28}$$

Dice and clDice are reported as percentages, Chamfer distance is reported in millimetres, and connected-component count is reported as a unitless count.

A.3.3 Comparison-Method Adaptation, Evaluation, and Timing

We adapted the available implementations of 3DGR-CAR (Fu et al., 2024), AutoCAR (Zhu et al., 2025), and DeepCA (Wang et al., 2025b) to the same ImageCAS-derived data protocol used by our

method. This experiment is not intended to reproduce the results reported on the original datasets of those papers. Instead, we train their models on our dataset and evaluate them under the same data split, input-view protocol, and metrics as our method to provide a controlled comparison. The available 3DGR-CAR and AutoCAR code did not contain every module required to train and evaluate their complete pipelines on our dataset. We therefore reimplemented the missing components from the algorithmic and implementation descriptions in the corresponding papers. The remaining adaptations principally concern the data-loading and evaluation interfaces. Each comparison-method loader reads the segmented two-dimensional coronary masks, their associated acquisition geometry, and the corresponding three-dimensional artery label, then converts these inputs to the tensor layout and spatial resolution expected by that method. The same training, validation, and test case assignments are used across methods. For each supported input-view count, all methods use the same selected views for a given test case so that differences are not caused by a different choice of projections.

Each comparison model is trained from scratch on our training split so that the experiment evaluates model capacity under the same data and supervision, without an advantage from external pretraining. Learning rate, training duration, loss weights, output threshold, and other method-specific settings are selected using reconstruction performance on the validation split. The test split is not used for model selection or hyperparameter tuning. Apart from modules needed to load our data or complete unavailable output and evaluation steps from the descriptions in the corresponding papers, the published model structure and learning objective are retained.

All final predictions are evaluated by the same code. Volumetric outputs are resampled to the common isotropic 0.5 mm grid and binarised using the validation-selected threshold before Dice and clDice are calculated. The native outputs have different spatial dimensions and voxel resolutions: AutoCAR produces a 400^3 volume at 0.5 mm resolution, whereas DeepCA and 3DGR-CAR produce 128^3 volumes at approximately 0.9167 mm and 0.9239 mm resolution, respectively. Consequently, resampling each prediction to the common grid can introduce method-dependent discretisation effects, so the Dice and clDice comparison is not entirely independent of native output resolution. Nevertheless, these dimensions are part of the original output configuration of each model; we retain them rather than altering the model architectures or retraining them for a common output grid. When a method does not natively produce an ordered centreline, its predicted volume is converted to a vascular graph using the same graph-extraction post-processing, and Chamfer distance is then computed between its extracted centreline point set and the ground-truth centreline point set in millimetres. Reported values are the mean and standard error across the test cases. This shared evaluator keeps the metric definitions, physical coordinate convention, and test cases fixed across all methods.

Inference-time breakdown. All runtimes in Table 8 use two input views and a single NVIDIA GeForce RTX 4090 GPU under the same software setup. We report *model inference time*, measured from supplying the 2D input views to obtaining the method's native reconstruction output; *post-processing time*, measured from the native output to a vascular graph/skeleton representation using the graph-extraction algorithm of AutoCAR (Zhu et al., 2025); and their sum as the *total time*. This post-processing is performed for every quantitatively evaluated case so that the final metrics use a graph/skeleton representation. Although the same graph-extraction algorithm is applied to all comparison methods, its runtime depends on the native input representation, evaluation-grid size, and vessel-mask complexity. DeepCA supplies a compact 128^3 binary `uint8` mask without probability thresholding and therefore has the shortest post-processing time. The 128^3 binary output of 3DGR-CAR is accompanied by additional arrays, requiring more input/output validation and potentially presenting a more complex foreground. AutoCAR usually supplies a substantially larger 400^3 probability volume, which incurs additional decompression, validation, and thresholding and consequently has the longest post-processing time.

A.3.4 LCA Reconstruction Results

Table 9 reports the corresponding LCA reconstruction results.

The LCA task is more challenging than RCA because its representation supports up to 13 branches, increasing anatomical complexity, projected branch overlap, and ambiguity under sparse views. This is reflected in the lower reconstruction accuracy of our LCA model relative to its RCA counter-

Table 8: Two-view reconstruction-time comparison on one NVIDIA GeForce RTX 4090 GPU. Model inference produces each method's native output, post-processing converts that output into a vascular graph/skeleton representation when required, and total time is their sum. The best result is shown in **bold** and the second-best result is underlined; post-processing times are ranked only among methods that require this stage. – denotes that post-processing is not required; N/R denotes that it was not recorded.

Method	Model inference time ↓	Post-processing time ↓	Total time ↓
SDF-CAR (Reda et al., 2026)	45 min	N/R	N/R
3DGR-CAR (Fu et al., 2024)	61 s	2.71 s	63.7 s
DeepCA (Wang et al., 2025b)	700 ms	1.99 s	2.69 s
AutoCAR (Zhu et al., 2025)	330 ms	3.43 s	3.76 s
Ours, coarse prediction	**60 ms**	–	**0.06 s**
Ours, refined prediction	121 ms	–	0.12 s

Table 9: LCA reconstruction from one, two, and four input views. CD denotes Chamfer distance, and CC count denotes connected-component count. Values are mean ± standard error. The best result is shown in **bold** and the second-best result is underlined. – denotes a view-count setting that is not reported in the corresponding paper.

Method	One view				Two views				Four views			
	Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓	Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓	Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓
3DGR-CAR (Fu et al., 2024)	**26.51** (±0.51)	**26.48** (±0.56)	**8.25** (±0.17)	46.43 (±3.54)	35.05 (±0.48)	37.16 (±0.54)	**5.87** (±0.11)	76.10 (±8.06)	43.64 (±0.66)	47.81 (±0.75)	**4.94** (±0.20)	100.88 (±11.59)
AutoCAR (Zhu et al., 2025)	–	–	–	–	46.89 (±1.29)	63.00 (±1.59)	6.13 (±0.38)	11.36 (±0.78)	–	–	–	–
DeepCA (Wang et al., 2025b)	–	–	–	–	**56.56** (±0.81)	**71.20** (±1.11)	6.66 (±0.24)	9.97 (±0.32)	–	–	–	–
Ours, coarse	11.15 (±0.76)	10.10 (±0.81)	14.08 (±0.51)	1.829 (±0.118)	14.92 (±0.77)	13.94 (±0.90)	12.37 (±0.51)	1.605 (±0.097)	14.59 (±0.73)	14.05 (±0.93)	12.46 (±0.49)	1.618 (±0.095)
Ours, refined	19.73 (±1.00)	21.36 (±1.19)	11.31 (±0.45)	**1.197** (±0.050)	49.45 (±1.32)	61.30 (±1.93)	5.98 (±0.47)	**1.053** (±0.026)	**53.45** (±1.28)	**67.43** (±1.93)	5.30 (±0.42)	**1.039** (±0.022)

part. In the conventional two-view setting, DeepCA achieves the highest Dice and clDice, whereas our method obtains a lower Chamfer distance than both AutoCAR and DeepCA and an average connected-component count of 1.053, substantially closer to the single connected ground-truth tree than their respective counts of 11.36 and 9.97 (AutoCAR and DeepCA, respectively). AutoCAR and DeepCA backproject the observed views into volumetric representations, which can favour agreement with the input silhouettes and voxel-overlap metrics. In contrast, the connected-component results and the qualitative examples in Figure 4 show that our explicit representation more consistently preserves a coherent artery-tree structure. Under one-view input, 3DGR-CAR obtains higher overlap scores but produces 46.43 connected components on average, compared with 1.197 for our refined prediction. Its predicted foreground volume is also substantially larger than the ground truth, indicating that part of its overlap arises from broad, fragmented coverage. Although a single projection does not determine depth, our model still produces a nearly connected artery-shaped 3D hypothesis rather than a fragmented volume. With four views, our method also improves Dice and clDice over 3DGR-CAR while retaining a connected prediction, although its Chamfer distance remains slightly higher.

A.3.5 Ablation Studies

With two input views, the branch-existence head achieves an accuracy of 96.30% and an F1 score of 98.12%, indicating reliable identification of the branches present in each case. Table 5 compares direct dense-centreline prediction with the B-spline parameterisation. The B-spline representation improves Dice, clDice, and Chamfer distance at both prediction stages for RCA and LCA. It also lowers the refined connected-component count for both arteries and the coarse count for LCA; the dense-centreline variant has a marginally lower RCA coarse count. Table 6 compares the frozen image encoders. After refinement, VGGT improves evaluation metrics over ResNet-101 for both

Table 10: Refinement ablation from a common two-view coarse initialisation. CD denotes Chamfer distance, and CC count denotes connected-component count. Values are mean ± standard error; post-coarse time is reported without standard error and excludes the shared coarse prediction. The best result is shown in **bold** and the second-best result is underlined; tied best results are both bold. Runtime ranking excludes the coarse row, which has no post-coarse processing.

Reconstruction stage	RCA				LCA				Post-coarse time ↓
	Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓	Dice (%) ↑	clDice (%) ↑	CD (mm) ↓	CC count ↓	
Coarse prediction	19.47 (±1.12)	20.40 (±1.56)	9.70 (±0.41)	1.324 (±0.064)	14.92 (±0.77)	13.94 (±0.90)	12.37 (±0.51)	1.605 (±0.097)	0
Direct per-scene optimisation	41.02 (±1.62)	50.02 (±2.24)	6.65 (±0.39)	1.243 (±0.057)	28.75 (±1.12)	33.16 (±1.51)	9.29 (±0.50)	1.421 (±0.073)	11 s
Learned geometry refiner	<u>65.89</u> (±1.44)	**80.84** (±1.90)	**3.16** (±0.27)	<u>1.068</u> (±0.029)	<u>48.43</u> (±1.29)	**61.38** (±1.93)	**5.98** (±0.47)	**1.053** (±0.026)	**14 ms**
Learned geometry and radius refiners	**66.92** (±1.50)	<u>80.62</u> (±1.92)	<u>3.16</u> (±0.27)	**1.054** (±0.026)	**49.45** (±1.32)	<u>61.30</u> (±1.93)	<u>5.98</u> (±0.47)	<u>1.053</u> (±0.026)	<u>61 ms</u>

RCA and LCA. At the coarse stage, VGGT improves Dice, clDice, and connected-component count for both arteries and Chamfer distance for LCA, while RCA Chamfer distance is slightly lower with ResNet-101. Table 7 evaluates within-case branch-subset augmentation. They improve evaluation metrics for both prediction stages and artery cohorts. These results support using the variants to learn optional side-branch structure. Table 8 separates native model inference from graph-extraction post-processing. Our native centreline-and-radius output requires no graph-conversion stage and completes refined two-view reconstruction in 0.12 s. Table 10 starts all alternatives from the same two-view coarse prediction. Direct per-scene optimisation improves upon the coarse initialisation, but remains worse than learned refinement in Dice, clDice, Chamfer distance, and connected-component count for both arteries. The optimiser minimises a projection loss against only the observed 2D masks; under sparse-view ambiguity, it can overfit projection-specific evidence without learning whether the resulting correction is plausible in 3D. In contrast, the refiner's residual updates are learned across training cases with 3D supervision, allowing it to use projection evidence while retaining learned artery structure. The geometry refiner therefore provides the dominant improvement over direct optimisation in milliseconds rather than seconds.