In the supplementary materials, we present details of the three datasets and the disconnected components search algorithm.

Table 3. Detailed properties for CTA Coronary datasets used in our study.

Dataset	No. of train validation, test	Modality	Coronary Artery Disease	In-plane resolution (mm)
ASCOS	28/4/8	3D CTA	Half with disease , half w/o disease	0.3-0.4
ImageCAS	140/20/40	3D CTA	All with disease	0.29-0.43
JHH	0/0/10	3D CTA	All with disease	0.35-0.40

Algorithm 1 Searching All Disconnection Point Detection for a CTA volume

Require: INPUT: $V \in \mathbb{R}^{D\times H\times W}$ // `3D CTA input volume`; $f(\cdot,\theta)$ // `repairing model with learned parameters` θ; N // `number of disconnected point`
Ensure: $P = \{P_1, ..., P_N\}$ // `predicted disconnected point coordinates`
1: Initialize: H $\leftarrow$ $initializeHeatmap(H) \in \mathbb{R}^{N\times D\times H\times W}$, $H = \{H_1, ..., H_N\}$ // `predicted disconnected point heatmaps`
2: Noise filter // `Excluding components smaller than 4 voxels`
3: $CandPts \leftarrow FindCandPts(V)$ // `Find candidate disconnected points (CandPts)`
4: $CandPts \leftarrow removeLargestCC(CandPts)$, $CandPts = \{c_1, ..., c_M\}$ // `Remove largest connected component (removeLargestCC)`
5: **for** $c \in CandPts$ **do**
6: $P'_c \leftarrow random\_sample(c) \in \mathbb{R}^3$ // `Randomly sample point from CandPts`
7: $\hat{H}_t \leftarrow f(P'_c, \theta)$ // `Generate heatmap for CandPts according to` P'_c
8: $location = getLocation(P'_c)$ // `Get the location in the heatmap grid`
9: $H[location] += \hat{H}_t$ // `Aggregate heatmaps`
10: **end for**
11: **for** $n = 1, ..., N$ **do**
12: $P_n \leftarrow \arg\max(H_n) \in \mathbb{R}^3$ // `Select highest intensity point`
13: **end for**
14: **return** $P = \{P_1, ..., P_N\}$ // `Return predicted disconnected points`