

# CPR-GCN: Conditional Partial-Residual Graph Convolutional Network in Automated Anatomical Labeling of Coronary Arteries

Han Yang<sup>b,\*</sup>, Xingjian Zhen<sup>§,\*</sup>, Ying Chi<sup>b,†</sup>, Lei Zhang<sup>b</sup>, and Xian-Sheng Hua<sup>b</sup>

<sup>b</sup>Alibaba Group, <sup>§</sup>UW-Madison

{xiuxian.yh, xinyi.cy, lei.zhang.lz, xiansheng.hxs}@alibaba-inc.com, {xzhen3}@wisc.edu

## Abstract

Automated anatomical labeling plays a vital role in coronary artery disease diagnosing procedure. The main challenge in this problem is the large individual variability inherited in human anatomy. Existing methods usually rely on the position information and the prior knowledge of the topology of the coronary artery tree, which may lead to unsatisfactory performance when the main branches are confusing. Motivated by the wide application of the graph neural network in structured data, in this paper, we propose a conditional partial-residual graph convolutional network (CPR-GCN), which takes both position and CT image into consideration, since CT image contains abundant information such as branch size and spanning direction. Two majority parts, a Partial-Residual GCN and a conditions extractor, are included in CPR-GCN. The conditions extractor is a hybrid model containing the 3D CNN and the LSTM, which can extract 3D spatial image features along the branches. On the technical side, the Partial-Residual GCN takes the position features of the branches, with the 3D spatial image features as conditions, to predict the label for each branches. While on the mathematical side, our approach twists the partial differential equation (PDE) into the graph modeling. A dataset with 511 subjects is collected from the clinic and annotated by two experts with a two-phase annotation process. According to the five-fold cross-validation, our CPR-GCN yields 95.8% meanRecall, 95.4% meanPrecision and 0.955 meanF1, which outperforms state-of-the-art approaches.

## 1. Introduction

Cardiovascular disease is one of the leading causes of death worldwide [21]. Cardiac CT angiography (CCTA) image is widely adopted for the diagnosis of the cardiovascular disease because of its non-invasion and high sensitiv-

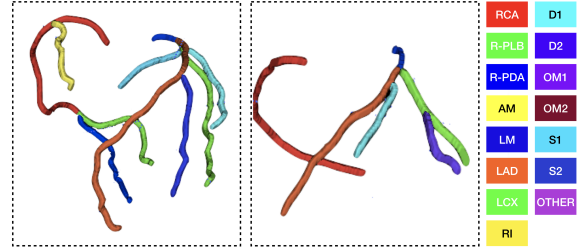


Figure 1. The examples of coronary arteries for different subjects. Every color indicates a specific coronary artery name. The number of branches, the direction, and the connection all vary between these two subjects. The left vessel tree containing 10 branches is more complete, while multiple vessel branches are missing in the right one.

ity [15]. In the clinic, doctors need a series of manual labor to obtain the diagnostic report, which is a time-consuming effort. If a computer-aided diagnosis (CAD) system can generate the diagnostic report automatically, a huge amount of time can be saved. While automated anatomical labeling of the coronary artery tree extracted from CCTA image is a prerequisite step in the automated CAD system.

The coronary artery tree consists of two components, i.e., left domain (LD) and right domain (RD). Also they both originate from the aorta. According to [23], the main coronary arteries of interest are left main (LM), left descending artery (LAD), left circumflex artery (LCX), left ramus-intermedius (RI), obtuse margin (OM), diagonal artery (D), septal artery (S), right coronary artery (RCA), right posterior lateral branches (R-PLB), right posterior descending artery (R-PDA), right acute marginal artery (AM) (shown in Figure 1). From prior knowledge, we know that LAD, LCX, and RI are from LM. R-PLB, R-PDA, and AM are from RCA. Also S's and D's are from LAD, while OM's are from LCX. This makes the whole coronary artery tree as structured data. Normally, RCA, LM, LAD, and LCX are treated as the main branches. Other branches are treated as the side branches.

Several anatomical labeling techniques have been developed for coronary arteries [23, 3, 22], brain arteries [1],

\*Equal Contribution.

†Corresponding Author.

abdominal arteries [18, 25] and airways [10]. However, as shown in Figure 1, the coronary arteries vary much among subjects, which is the main challenge for the labeling system. The number of branches, the length and size of each branch, and the direction that the branch span all vary from person to person. [23, 3] both rely on registration algorithm and prior knowledge. They first identified four main branches (i.e., LM, LAD, LCX and RCA) and then labeled the side branches (e.g., AM, R-PDA, D, etc.). Finally, the results were refined by logical rules which are translated from the clinical experience. But these conventional methods are not data-driven, i.e., they can not leverage the advantage of big data. Recently, in [22], the author brought a novel deep neural network (TreeLab-Net), which can learn from the position features extracted from the coronary artery centerlines for labeling the segments. Although, this model is data-driven, it only utilizes position information from vessel centerline and leaves the full information in the CCTA images aside. In addition, the input for the TreeLab-Net is built by topological structure. Missing the main branches (e.g., LM, RCA, etc.) might have deep influence in labeling the side branches. Therefore, a robust and self-adaptive model is needed for this structured data.

In the deep learning field, it's quite mature for the study in Euclidean space where elements are treated equally. However, there are normally two view aspects for the structured data. In [5, 4], the authors viewed the structured data from a manifold-valued aspect. Also in [14, 12, 26] where the structured data can be viewed as the graph, the authors introduced the graph models with nodes and edges, which can be used to extract the information from the relationship between each node in the structured graph data. It's natural to treat the coronary arteries as the tree because of the path the branches spread and the connection among the branches.

In this paper, we propose a conditional partial-residual graph convolutional network (CPR-GCN), which can make full use of both position information and 3D image information in the CCTA volume. The partial-residual block is applied to the position domain features to enhance the features. Also, we use 3D Convolutional Neural Network (CNN) together with Bidirectional Long Short-Term Memory (BiLSTM) to extract the features along each branches as the conditions for the graph model. These two parts compose the CPR-GCN which can be trained end-to-end.

In a summary, our main contributions are as follows:

- We propose the CPR-GCN, a conditional partial-residual graph convolutional network, which can label the coronary artery tree end-to-end.
- To our best knowledge, this is the first time we take 3D image features into consideration in coronary artery labeling field.
- Our CPR-GCN and the hybrid model (i.e., 3D CNNs

following BiLSTMs) can be jointly trained. We evaluate the CPR-GCN on a large private collected dataset. Our approach outperforms the state-of-the-art result.

## 2. Related Work

Most of the related work can be divided into two categories, i.e., traditional based and deep learning based. Traditional methods are based on the knowledge and the topology of the coronary arteries. Normally, they require two steps of registration and correction. With the development of deep learning, there are also some methods on the graph-based structural data. These methods extract features as nodes and train the model with the data they acquire. Also these methods heavily depend on the size and quality of the dataset.

### 2.1. Traditional Methods

Most traditional methods are based on registration. In [23], the author presented a two-step method. In the registration step, the main branches, LM, LAD, LCX, and RCA, are identified. Then the rest branches are matched afterward. In [3], they built the 3D models for both right dominant and left dominant. The 3D coronary trees from subjects are aligned with the 3D models to get the label of each segment. They also applied the logical rules to fulfill the clinical experience.

However, traditional methods highly rely on the main branches. If the main branches are missing in the automated segmentation system, the performance will deteriorate dramatically. Also, they require prior knowledge about how the coronary tree span. Thus if some sub-branches are missing, it will affect the topology and the performance. Finally, all these traditional methods have the human-interpretation, which means that if the topology is too complicated, the automatic system will raise the information that it is not capable of determining.

### 2.2. Deep Learning Based Methods

With the power of deep learning, the TreeLab-Net is developed in [22]. They combined the multi-layer perceptron encoder network and the bidirectional tree-structural LSTM to construct the TreeLab-Net. They used several selected features from positions and stacked the Child-Sum Tree LSTMs as the components. The left and right coronary arteries are trained independently.

In this method, the missing branches will cause a massive problem since this will change the layer index of the nodes inside the tree. With the tree-structured model, the message is only passing between nearby layers. The closer the branches are to the root nodes, the higher impact they will raise. The left and right are also classified with the rules and the thresholds. This method has a strong assumption that the branches will only bifurcate so that each node in the

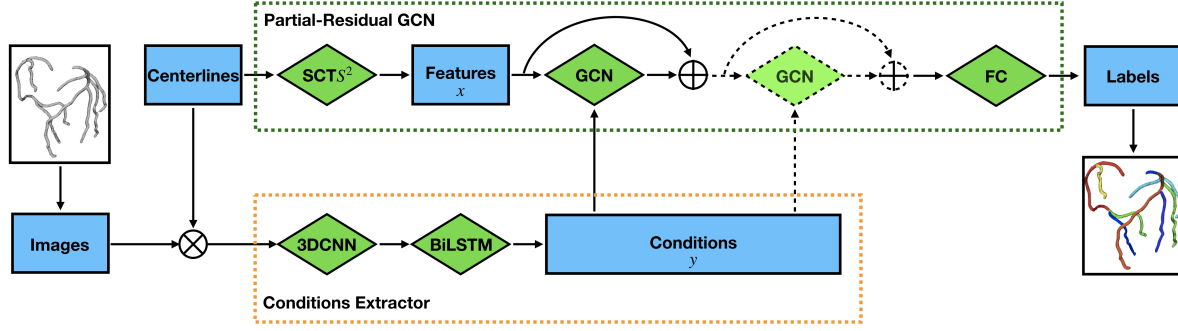


Figure 2. The framework for the CPR-GCN. The two parts, conditions extractor and partial-residual GCN, compose our CPR-GCN model. The backbone of our model is in the green bounding box. The orange bounding box extracts extra information from image domain. The  $\oplus$  is the residual connection block on the features  $x$ . The  $\otimes$  means using control points along the centerlines to extract the moving cubes from CCTA images.

tree-structure can only have two children. But in the coronary arteries, it's quite reasonable that several sub-branches bifurcate from the points near to each other. So the node for the parent branch will return to have more than two children, which is beyond the capacity of the TreeLab-Net.

In a broad sense, the TreeLab-Net is a simplified version of the graph models. In [14], the author brought out the Graph Convolutional Networks (GCN) which operate directly on graphs. By projecting the graphs into the Fourier domain, the author defined the convolution operator and filter kernels in the Fourier domain using the Chebyshev polynomials. In [17], they modified the Graph Neural Networks (GNN) [19] to use gated recurrent units.

Even though these graph models are successful in molecular fingerprints [7] and protein interface prediction [8], they have not been used in labeling coronary arteries. Also graph models also suffer from shallow structure problems. Since stacking multiple GCN layers will result in over-smoothing [16].

### 3. Our Approach

In this section, we detail the CPR-GCN, which makes full use of both CCTA images and the position of the coronary artery centerlines. As shown in Figure 2, our method considers both CCTA images and the features from positions mentioned in [22]. The centerlines are extracted using the automated coronary artery tracking system [24]. Then, our CPR-GCN extracts the features from the centerlines within  $SCTS^2$  block. Also, in the image domain, we use the sub-sampled control points on centerlines to get the moving cubes with a fixed radius  $\gamma$  along each branches as the image domain data. The conditions for our CPR-GCN model are obtained with the 3D CNN and the BiLSTM. The detailed architecture of the CPR-GCN model is shown in Table 1.

Table 1. The details of the parameters in our model. The framework is shown in Figure 2.

Block	Details
$SCTS^2$	first, middle, and last points tangent direction and first-last direction
3D CNN	kernel size = 3, in channel = 1, out channel = 16 maxpooling size = 2 kernel size = 3, in channel = 16, out channel = 32 maxpooling size = 2 kernel size = 3, in channel = 32, out channel = 64 maxpooling size = 2
BiLSTM	layer = 4, hidden size = 128
CPR-GCN	out channel = 256 out channel = 256 out channel = 256
Fully Connected	out channel = 128 out channel = # of classes

#### 3.1. Position Domain Features

In [22], the author introduced a spherical coordinate transform 2D (SCT2D) which transforms the positions  $P_k = [(x_i, y_i, z_i)]_{i=1}^{Length_k}$  in 3D into the azimuth and elevation angles  $[(\varphi_i, \theta_i)]_{i=1}^{Length_k}$ . They argued that this could normalize the variance of centerlines in the default Cartesian coordinate system. However, it is noticeable that  $\varphi$  and  $\theta$  have the periodic  $2\pi$ . So with limiting the range of the angle to be  $[0, 2\pi)$ , the small amount of shaking, due to the noise, will return to be  $2\pi$  difference near the angle 0.

The similar idea of using spherical coordinate transform is applied in our approach. Since each branches are processed separately, so as to get the azimuth and elevation angles, we need to define the origin and the  $x, y, z$  axes for each branches. For each branches, the first control point is chosen as the origin. The direction pointing from the first point to the second point is defined as  $z$  axis. The vector from the first point to the last point of the centerline lies in  $y - z$  plane.

To overcome the instability due to the periodic, we use the  $S^2$  manifold to represent  $\varphi, \theta$ . The  $S^2$  manifold is the

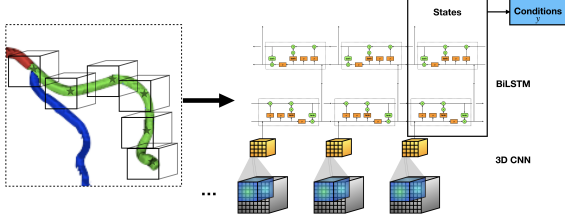


Figure 3. The details of the framework in the image domain. We use 3D CNN and the BiLSTM to learn the conditions in the image domain for our partial-residual block. We use the last state as the final representative of the conditions.

sphere with unit radius in  $\mathbb{R}^3$ . A trivial method is to use the  $2 \times 2$  matrix  $M = \begin{bmatrix} \sin\theta & \sin\varphi \\ \cos\theta & \cos\varphi \end{bmatrix}$ . Because of the periodic of  $\sin(\cdot)$  and  $\cos(\cdot)$ , the matrix  $M$  is stable on the whole manifold  $S^2$ . This kind of spherical coordinate transformation is called SCTS $^2$  in the rest of this paper. The Cartesian coordinate and the  $S^2$  manifold transformation is in Eq. 1.

$$\begin{aligned} x &= r \sin\theta \cos\varphi & r &= \sqrt{x^2 + y^2 + z^2} \\ y &= r \sin\theta \sin\varphi & \cos\theta &= z/r, \theta \in [0, \pi] \\ z &= r \cos\theta & \sin\varphi &= x/(r \sin\theta), \cos\varphi = y/(r \sin\theta) \end{aligned} \quad (1)$$

We use the similar features mentioned in [22]: (1) The  $S^2$  projection and the normalized 3D positions of the first point, center point, and the last point. (2) Directional vector between first and last points and the tangential direction at the start point in both 3D and  $S^2$ .

### 3.2. Image Domain Conditions

Most of the medical images, including Magnetic resonance imaging (MRI) and the CCTA we use, lie in 3D. The branches, unlike other images, have the sequential dependency. Thus we use the 3D CNN to extract the spatial features and use the BiLSTM afterward to summarize the tubular sequential features. An example of processing the R-PLB branch is shown in Figure 3. The  $z$  dimension of CCTA images are the slices, which might have different spacing from  $x$  and  $y$  dimensions. So we resample all the CCTA images to have the same spacing  $v$  along  $x, y, z$  dimensions.

Using the automatic segmentation method, we can get the centerlines of all branches ( $P_1, P_2, \dots, P_n$ ) which build into a coronary artery tree. We separate the branches where the centerlines end or bifurcate. If the starting points of the children branches are close to each other, we consider all these children branches are from the same point on the parent branch. The control points of the centerlines are smoothed using the Catmull-Rom spline [20]. Finally, the control points are sub-sampled with the same length.

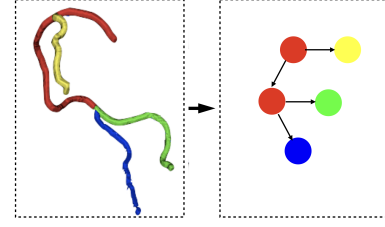


Figure 4. The rules we used to build the graph for the CPR-GCN. Whenever the branches bifurcate, new node is added in the graph. So main branches (e.g., RCA, LAD, etc.) might be represented by multiple nodes in the graph. For example, two red nodes in the right box belong to RCA.

The image domain data is the cubes  $I_k$  with the fixed radius  $\gamma$  around each control points of  $P_k$ . Three layers of 3D CNN and 3D maxpooling are used to extract the features of  $I_k$ . The weights of CNN are shared among the segments. In order to train the model in a mini-batch manner, these feature vectors are padded to the  $\max_{k=1}^n (\text{Length}_k)$  as the input of the multi-layer bidirectional LSTM [9]. The last hidden state is treated as the final conditions  $y$ , which represents the image information of this branch. We treat this as the conditional information since the images are of less importance than the position domain features.

### 3.3. Partial-residual Block of GCN

The layer-wise propagation rule for the traditional GCN is Eq. 2. In the multi-layer GCN, the features  $X$  of nodes is the input for the first layer,  $X \in \mathbb{R}^{n \times d_0}$ . Here, the  $n$  is the number of nodes. The  $d_0$  is the dimension of the features for each nodes. The  $A$  is the adjacency matrix for the graph. The  $W^l$  is the layer-wise trainable weights,  $W^l \in \mathbb{R}^{d_l \times d_{l+1}}$ . The  $\sigma(\cdot)$  is the activation function. In this paper, we choose  $\text{ReLU}(\cdot) = \max(0, \cdot)$  as our activation function to include the nonlinear ability.

$$\begin{aligned} H^{(l+1)} &= \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^l W^l) \\ \tilde{A} &= A + I_N, \tilde{D}_{ii} = \sum_j \tilde{A}_{ij} \end{aligned} \quad (2)$$

The  $\tilde{A}$  is the adjacency matrix  $A$  added the self-loop identity matrix  $I_N$ . The input for the first layer is  $H^0 = X$ .

Our conditional partial-residual block requires both the position features  $x$  and the CCTA image domain conditions  $y$ . The combination of the features and conditions from two domains is used as the representative of the nodes in the graph model. The edges are defined as the parent-children relationship. As we mentioned above, the topology of the whole coronary tree is collected from  $(P_1, P_2, \dots, P_n)$ . Whenever the branches bifurcate, we treat these parent-children branches as three (or more) nodes. The edges are from the parent branch to the children branches. This will build the graph of the subject with the adjacency matrix

A. As shown in Figure 4, the RCA first bifurcates the AM and then bifurcates the R-PLB and R-PDA. So the extracted graph has 4 edges and 5 nodes.

In [11], the author argued that the deep residual learning framework (Figure 5 (a)) can help in the performance. Instead of directly learning the map  $\mathcal{H}(\mathbf{x})$ , the neural network learns the residual part  $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$  assuming the input and the output having the same dimension. When changing the output dimension, it's a straightforward method to add the linear projection  $W_s$  by the shortcut connection:

$$\mathbf{z} = \mathcal{F}(\mathbf{x}, W_i) + W_s \mathbf{x} \quad (3)$$

Also in [2], the authors extended the idea of residual connection into the Residual Gated Graph ConvNets. The updated propagation rule is:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^l W^l) + H^l \quad (4)$$

If we view the residual connection in Eq. 4 as a continuous function of  $l$  and add enough number of layers. In the limit, [6] parameterize the continuous dynamics of hidden units using the ordinary differential equation (ODE):

$$\frac{dH(l)}{dl} = f_A(H(l), W(l), l) \quad (5)$$

In our setup, we have two kinds of input: position domain features  $x(l)$  and the CCTA image domain conditions  $y$ . If we treat  $l$  as the layer index number and  $x$  is the function of layer  $l$  in Eq. 5, we have the partial differential equation (PDE) on  $x, y$ :

$$\nabla H^l(x(l), y) = \frac{\partial H^l(x, y)}{\partial x} dx + \frac{\partial H^l(x, y)}{\partial y} dy \quad (6)$$

$$= \frac{\partial H^l(x, y)}{\partial x} \frac{dx}{dl} dl \quad (7)$$

$$= \text{GCN}_A(x, y) dl \quad (8)$$

Eq. 7 is based on the fact that we treat  $y$  as the conditions. We use the trainable  $\text{GCN}_A(x, y)$  to approach the partial

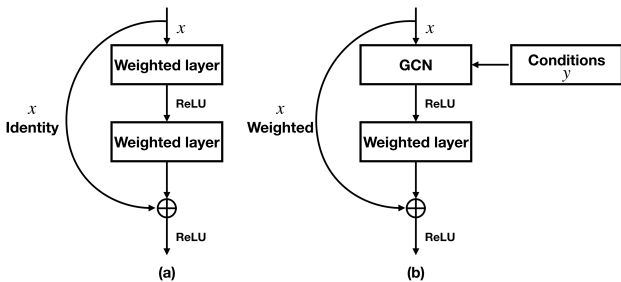


Figure 5. The partial-residual block in the CPR-GCN. It can strengthen some part of the features to have more influence on the final layer while absorbing the other as the conditions. (a) is the traditional residual block, (b) is our partial-residual.

differential  $\frac{\partial H^l(x, y)}{\partial x} \frac{dx}{dl}$ . If we take  $H^0 = x(0) = X$  and approximate  $dl = 1$ , we have

$$\begin{aligned} H^1 &= \nabla H^l(x, y) dl + H^0 \\ &= \text{GCN}_A(x, y) + X \end{aligned} \quad (9)$$

as the discrete numerical estimates. In our case, as shown in Figure 5 (b), the discrete partial-residual block of the CPR-GCN takes the weighted  $X$  because of the change of the channel sizes. If we push our PDE a bit further, we can have:

$$\begin{aligned} H^k &= \int \nabla H^l(x, y) dx + H^0 \\ &= \int_0^k \text{GCN}_A(x, y) dl + X \end{aligned} \quad (10)$$

Also, the  $X$  here should be weighted for the flexibility of the channels.

### 3.4. Data Flow

The algorithm of our CPR-GCN is shown in Alg. 1. CCTA image  $I$ , centerlines  $P_1, P_2, \dots, P_n$  and reference labels  $GT_1, GT_2, \dots, GT_n$  compose the training sample of the model. We first build the graph  $A$  via the bifurcation of  $P_1, P_2, \dots, P_n$ . The position domain feature  $x$  and the image domain feature  $y$  are extracted via SCTS<sup>2</sup> and a hybrid network (i.e., 3D CNN following BiLSTMs). In addition, we concatenate  $x$  and  $y$  as the input of GCN layers with  $x$  as the shortcut in residual connection. At last, a fully connected layer predicts the final labeling  $L_1, L_2, \dots, L_n$  and our object is to minimize the cross-entropy of these two distributions, i.e.,  $GT_1, GT_2, \dots, GT_n$  and  $L_1, L_2, \dots, L_n$ .

---

#### Algorithm 1: Training procedure of our approach

---

**Data:** CCTA image ( $I$ ), centerlines ( $P_1, P_2, \dots, P_n$ )  
**Ground truth:**  $GT_1, GT_2, \dots, GT_n$   
 $A \leftarrow \text{Build\_Graph}(P_1, P_2, \dots, P_n);$   
 $x_1, x_2, \dots, x_n \leftarrow \text{SCTS}^2(P_1, P_2, \dots, P_n);$   
 $I_1, I_2, \dots, I_n \leftarrow I \otimes (P_1, P_2, \dots, P_n);$   
 $y_1, y_2, \dots, y_n \leftarrow \text{BiLSTM} \leftarrow \text{3DCNN}(I_1, I_2, \dots, I_n);$   
 $H \leftarrow \text{GCN}_A(x, y) + x;$   
 $L_1, L_2, \dots, L_n \leftarrow \text{FC}(H);$   
 $\text{loss} \leftarrow \text{Cross\_Entropy}(L, GT);$

---

## 4. Experimental Results

### 4.1. Dataset and Evaluation Metrics

To our best knowledge, a public dataset with CTA image and annotation of coronary artery labeling is not available until now. Previous works [23, 3, 22] all collected a private experimental dataset from clinic. For instance, conventional methods [23, 3] only used 58 and 83 subjects respectively,

while deep learning based method [22] used a larger dataset with 436 subjects. In this study, we collected the largest relevant dataset from clinic. All vessel centerlines are first extracted using [24]. This dataset contains 511 subjects and all of them are annotated by two experts with a two-phase annotation process. These two experts give a label to every branch alone in the first round. Then the annotation results are merged and experts take discussions on inconsistent ones to obtain a final label. These 511 subjects and corresponding annotation compose our experimental dataset. The average number of branches in each subject is 9.65, with standard deviation (std) 2.13. The largest number of branches is 15, and the smallest number is 3. After bifurcated and separated, the average number of segments is 13.23. The detail is mentioned in Table 2. The edges in the table represent the relationship in the graph we build for each subject. The average number of edges in each graph is 11.18.

The evaluation is performed on all branch segments by the predicted label and the ground truth label. The recall rate for each segments is calculated by  $\text{Recall} = \frac{t_p}{t_p + f_p} |(\text{label} = i)$ . The precision is  $\text{Precision} = \frac{t_p}{t_p + f_n} |(\text{label} = i)$ . The F1 score is  $\text{F1} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} |(\text{label} = i)$ . Since the numbers of segments in are imbalance, we also use the mean metrics of all classes for the segments.  $\text{meanRecall} = \frac{1}{n} \sum_{i=1}^n \text{Recall}$ . It is similar for the other metrics.

## 4.2. Implementation Details

**Hyper-parameters selection** The CCTA images are scaled to  $v = 0.5$  mm voxel spacing. Since the radius of branches usually ranges in (0, 3) mm, i.e., (0, 6) voxels. So the radius of the cube is chosen to be  $\gamma = 12$  voxels to keep the angle, size, and the texture information. Thus the subsampling rate for the centerline positions is 10 voxels to have overlapping as well as keep the sequential information along the branches.

**Training** The dataset is randomly and equally divided into five subsets. In the training stage, we use a five-fold cross-validation strategy to evaluate all subjects in the dataset. The proposed CPR-GCN model has two trainable components, i.e., 3D CNN following LSTMs (3D CNN module), GCNs following a FC layer (GCN module). A series of 3D image cubes extracted along the vessel centerlines from the

Table 2. The basic information for the dataset we use. Segments are the branches after bifurcating. The edges are the relationship between segments.

	Number	Avg.(std))	Max	Min
Branches	4929	9.65 (2.13)	15	3
Segments	6760	13.23 (3.55)	22	3
Edges	5714	11.18 (3.56)	20	2

3D CCTA image are the input of the 3D CNN module. The position domain features extracted from the vessel tree via SCTS<sup>2</sup> are concatenated with the output of the 3D CNN module. The GCN module takes these combined features as input and predicts the label for every segment. In addition, the reference labels are needed to compute the cross-entropy loss with the predicted labels. The 3D image cubes, position domain features and the reference labels compose the training samples. Our CPR-GCN model is trained in an end-to-end manner.

The algorithm is implemented using Pytorch with an NVIDIA Tesla P100 GPU. We use Adam optimizer [13] with an initial learning rate of 0.001. Each mini-batch contains 8 coronary artery trees. For each training period, we train the CPR-GCN model up to 200 epochs which takes 2.7 hours. So the total training time for five-fold cross-validation is 13.5 hours.

**Testing** We first choose the best model in each fold according to the overall testing precision ignored classes. Then we use each model to evaluate the corresponding testing data. During inference, the average time spent on the CPR-GCN model is 0.045 s per case, which is greatly important in the clinical utilization.

**Results** Since there is not a public dataset in this field and conventional methods only evaluated their performance on a small private dataset. We also reproduced the conventional method and the deep learning based TreeLab-Net. Considering conventional methods [23, 3] mainly rely on registration and prior knowledge, we only reproduce [3] which has a improvement on [23]. Table 3 reports the detail performance on our dataset. Our CPR-GCN achieves the highest meanRecall of 0.958, meanPrecision of 0.954 and meanF1 of 0.955, which outperforms other methods with a large margin.

All the models have performed well on the main branches. But the side branches are also a crucially important part of the automated anatomical labeling in the CAD system. In our approach, we treat the main branches and side branches equally. So compared with other two-step methods, most of the side branches, such as OM and R-PDA, performs better in our approach. But since the number of segments of the main branches like LM and RCA is relatively large, the performance of these main branches is better. For the side branches, especially for the D and OM, the number of samples in the dataset is relatively small. So the performance is slightly worse than the main branches.

## 4.3. Ablation Study

To make sure that all the components of the CPR-GCN perform well, we design the ablation study experiments.

**Image domain condition** One of the major difference between our approach and the other methods is that we use the



Table 3. Comparisons of conventional method [3], deep learning based TreeLab-Net [22] and our CPR-GCN on our dataset. Recall, precision and F1 score are used as the evaluation metrics.

Method	Metric	RCA	R-PDA	R-PLB	AM	LM	LAD	LCX	RI	D	OM	S	Avg(std)
Conventional [3]	Recall	0.918	0.850	0.852	0.893	0.984	0.911	0.832	0.848	0.799	0.720	0.835	0.859±0.066
	Precision	0.925	0.835	0.860	0.871	0.991	0.929	0.810	0.803	0.781	0.739	0.865	0.855±0.069
	F1	0.922	0.842	0.856	0.882	0.987	0.920	0.821	0.825	0.789	0.730	0.850	0.857±0.067
TreeLab-Net [22]	Recall	0.950	0.858	0.818	0.871	0.996	0.948	0.913	0.770	0.816	0.805	0.862	0.873±0.067
	Precision	0.948	0.823	0.842	0.871	0.970	0.937	0.0.936	0.714	0.841	0.807	0.859	0.868±0.072
	F1	0.949	0.840	0.830	0.871	0.983	0.942	0.924	0.741	0.829	0.807	0.860	0.871±0.069
<b>Our CPR-GCN</b>	Recall	0.994	0.930	0.944	0.991	0.994	0.990	0.982	0.921	0.936	0.896	0.954	<b>0.958±0.033</b>
	Precision	0.987	0.946	0.947	0.983	0.984	0.986	0.971	0.896	0.883	0.933	0.974	<b>0.954±0.035</b>
	F1	0.990	0.938	0.945	0.987	0.989	0.988	0.976	0.909	0.909	0.914	0.964	<b>0.955±0.032</b>

extra information from CCTA images domain. As shown in the second column of Table 5, if we only remove the image domain conditions, the metrics, i.e., meanRecall, meanPrecision and meanF1 will drop over 2.0%. For instance, the meanF1 score is 0.934 compared with 0.955 in our CPR-GCN.

**Residual GCN connection** Also, our approach brought the partial-residual connection in the graph model. In this part, we only remove the residual connection in the GCN block and keep other setting the same as the CPR-GCN. The third column in Table 5 reports that the F1 score will drop to 0.947. This argues that with the help from residual, our model can absorb the features from both position and image domain as well as keep the original position domain features.

**Undirected graph** In this part, we build the undirected graph, which means adding the opposite edges from the original graph. Detailed results are illustrated in Table 5, the fourth column. Although the average metrics among classes (i.e., meanRecall, meanPrecision and meanF1) are slightly worse than our best result, several classes (e.g., LM, LAD, R-PDA, etc.) for undirected graph have higher precision than directed graph. It is worth to note that we select directed graph to achieve higher average metrics. In clinical practice, we can make choice according to the requirement

of the doctors.

**Repeated GCN Blocks** [16] reports that stacking multiple GCN layers will result in over-smoothing. We also conduct the experiment to answer how the number of GCN blocks influences the performance. Considering the connection complexity in coronary tree graph, we respectively evaluate our CPR-GCN with 1, 2, 3, 4 GCN layers. Figure 6 illustrates that we can improve the performance by stacking GCN blocks, especially from 1 GCN block to 2 GCN blocks. However, the model with 4 GCN blocks has no obvious improvement compared with 3 GCN blocks. Therefore we use 3 GCN blocks in our CPR-GCN. The depth of the graph for the coronary artery tree mostly is less than 4, which might lead to the evaluation results.

#### 4.4. Synthetic "Data Attack"

We hold the opinion that our CPR-GCN is more robust when main branch in the vessel tree is missing. So we first build a synthetic dataset from our original dataset. 20% LM and RCA branches is randomly removed. Most of the other side branches (e.g., LCX, LAD, RI, AM, etc.) directly originate from those two branches. In this new sythetic dataset, 295 RCA and LM branches are removed. 1123 of 6760 vessel segments directly connect with these 295 missing branches. Since conventional methods [23, 3] strictly rely on the main branches. We only evaluate the trained CPR-GCN and TreeLab-Net [22] on this synthetic dataset. As shown in Table 4, our CPR-GCN drops alomost 2.6% while the TreeLab-Net drops almost 6.7% in three average metrics. This demonstrates that our method is more robust.

Table 4. Experimental results of TreeLab-Net [22] and our CPR-GCN on the original dataset and synthetic dataset. 20% RCA and LM is randomly removed in the synthetic dataset.

Method	Dataset	meanRecall	meanPrecision	meanF1
TreeLab-Net [22]	Original	0.873	0.868	0.871
	Synthetic	0.806	0.799	0.802
<b>Our CPR-GCN</b>	Original	0.958	0.954	0.955
	Synthetic	0.931	0.928	0.929

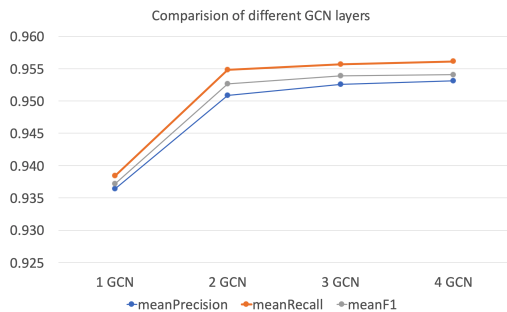


Figure 6. Ablation study of repeated GCN blocks. 1 GCN, 2 GCN, 3 GCN and 4 GCN represent stacking corresponding number of the GCN block

Table 5. Part of the ablation study results for our approach. The image domain conditions, as well as the partial-residual connection, are both essential parts of the CPR-GCN.

Ablation Study	Without image domain conditions			Without residual connection			Undirected graph			Our CPR-GCN		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
LM	0.990	0.994	0.992	0.959	0.984	0.971	0.996	0.998	0.997	0.984	0.994	0.989
LAD	0.977	0.983	0.980	0.975	0.978	0.976	0.987	0.993	0.990	0.986	0.990	0.988
LCX	0.938	0.964	0.951	0.946	0.963	0.955	0.963	0.980	0.971	0.971	0.982	0.977
RI	0.880	0.904	0.892	0.917	0.871	0.893	0.875	0.904	0.890	0.896	0.921	0.909
RCA	0.982	0.989	0.986	0.980	0.981	0.980	0.989	0.994	0.992	0.987	0.994	0.991
D	0.842	0.877	0.859	0.883	0.934	0.908	0.889	0.924	0.906	0.883	0.936	0.909
S	0.954	0.954	0.954	0.979	0.948	0.963	0.967	0.937	0.952	0.974	0.954	0.964
OM	0.830	0.814	0.822	0.915	0.913	0.914	0.912	0.904	0.908	0.933	0.896	0.914
R-PDA	0.938	0.927	0.933	0.952	0.950	0.951	0.960	0.939	0.949	0.947	0.944	0.945
R-PLB	0.925	0.925	0.925	0.951	0.930	0.941	0.947	0.953	0.950	0.946	0.930	0.938
AM	0.977	0.977	0.977	0.968	0.971	0.969	0.988	0.997	0.993	0.983	0.994	0.987
<b>Avg.</b>	0.930	0.937	0.934	0.948	0.947	0.947	0.952	0.957	0.954	<b>0.954</b>	<b>0.958</b>	<b>0.955</b>
<b>Std.</b>	0.054	0.053	0.053	0.037	0.036	0.035	0.038	0.036	0.036	<b>0.035</b>	<b>0.033</b>	<b>0.032</b>

## 5. Discussion

As shown above, our approach achieves state-of-the-art result. The CPR-GCN takes the image domain information from 3D CCTA images as the conditions for our approach. So as to stress the importance of the features from position domain, we introduced the partial-residual block on the position domain features.

**Image domain information** As far as we know, we are the first to include the image domain information as the conditions. The result, which is shown above, suggests that even though the position domain features are important components of the automated anatomical labeling of coronary arteries, the CCTA images weigh more than just for semantic segmentation. For example, the sizes and the shrinking points are not visible in the centerline-based position features, which may be extracted from the images. So even though the positions themselves have relatively rich information in this problem, all the other methods lack the ability to absorb the information from the original 3D CCTA image.

**Partial-residual block** One of the main contributions of our approach is that we brought the partial-residual block. In the traditional residual block, all dimensions of the input  $x$  are treated equally. But in the coronary arteries labeling problem, the positions are proven to play an important role. So as to stress this importance as well as use the extra information inside the CCTA images, we use the partial-residual block to treat the image domain in formations as the extra conditions of the model. With the ablation study, we notice that this kind of structure can improve the metric from 0.947 to 0.955. Also, it can make the model more stable.

**Robustness** Our CPR-GCN is purely driven by data. So the CPR-GCN can make use of the enormous size of data.

Without prior knowledge and the hard-coded “rule”, the CPR-GCN is more robust to the noises. All the nodes in our graph, which represent different segments, weigh the same in the CPR-GCN. So the wrong classification of LM or other major branches has less chance to spread through all other branches. In order to prove this opinion, we conduct a interesting synthetic “Data Attack” experiment. The results show that missing main branch has less impact on our CPR-GCN than other deep leaning method.

**Disadvantages and future work** There are still some future works for this problem. Since we treat every branches equally, the imbalance among branches is an issue. It’s noticeable that the main branches perform better than side branches due to the number of samples are different. There will also be some tiny branches missing after the segmentation, which will increase the imbalance. Also in this model, we use the discrete approximation of the PDE. In the future, the model can be pushed forward to the continuous model.

## 6. Conclusion

In this paper, we propose the end-to-end conditional partial-residual graph convolutional network (CPR-GCN) model for automated anatomical labeling of coronary arteries, where few alternatives are available. Compared with the traditional methods and the recent deep learning based methods, our approach achieves the state-of-the-art result. We show that with the conditional partial-residual block, both information in position domain and CCTA image domain can be taken into consideration. On the experiment side, we show that our CPR-GCN is more robust and flexible compared with others. The result also shows that the CCTA image domain matters in coronary arteries labeling. Importantly, we show that our algorithmic contributions facilitate the CAD system.



## References

- [1] Hrvoje Bogunović, José María Pozo, Rubén Cárdenes, Luis San Román, and Alejandro F Frangi. Anatomical labeling of the circle of willis using maximum a posteriori probability estimation. *IEEE transactions on medical imaging*, 32(9):1587–1599, 2013.
- [2] Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.
- [3] Qing Cao, Alexander Broersen, Michiel A de Graaf, Pieter H Kitslaar, Guanyu Yang, Arthur J Scholte, Boudewijn PF Lelieveldt, Johan HC Reiber, and Jouke Dijkstra. Automatic identification of coronary tree anatomy in coronary computed tomography angiography. *The international journal of cardiovascular imaging*, 33(11):1809–1819, 2017.
- [4] Rudrasis Chakraborty, Jose Bouza, Jonathan Manton, and Baba C Vemuri. A deep neural network for manifold-valued data with applications to neuroimaging. In *International Conference on Information Processing in Medical Imaging*, pages 112–124. Springer, 2019.
- [5] Rudrasis Chakraborty, Chun-Hao Yang, Xingjian Zhen, Monami Banerjee, Derek Archer, David Vaillancourt, Vikas Singh, and Baba Vemuri. A statistical recurrent model on the manifold of symmetric positive definite matrices. In *Advances in Neural Information Processing Systems*, pages 8883–8894, 2018.
- [6] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.
- [7] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.
- [8] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. In *Advances in Neural Information Processing Systems*, pages 6530–6539, 2017.
- [9] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.
- [10] Suicheng Gu, Zhimin Wang, Jill M Siegfried, David Wilson, William L Bigbee, and Jiantao Pu. Automated lobe-based airway labeling. *Journal of Biomedical Imaging*, 2012:1, 2012.
- [11] 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*, pages 770–778, 2016.
- [12] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.
- [13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [15] Jonathon Leipsic, Suhny Abbbara, Stephan Achenbach, Riccardo Cury, James P Earls, GB John Mancini, Koen Nieman, Gianluca Pontone, and Gilbert L Raff. Sctt guidelines for the interpretation and reporting of coronary ct angiography: a report of the society of cardiovascular computed tomography guidelines committee. *Journal of cardiovascular computed tomography*, 8(5):342–358, 2014.
- [16] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [17] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.
- [18] Tetsuro Matsuzaki, Masahiro Oda, Takayuki Kitasaka, Yuichiro Hayashi, Kazunari Misawa, and Kensaku Mori. Automated anatomical labeling of abdominal arteries and hepatic portal system extracted from abdominal ct volumes. *Medical image analysis*, 20(1):152–161, 2015.
- [19] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.
- [20] Christopher Twigg. Catmull-rom splines. *Computer*, 41(6):4–6, 2003.
- [21] World Health Organization (WHO). The top ten causes of death - fact sheet n310. 2008.
- [22] Dan Wu, Xin Wang, Junjie Bai, Xiaoyang Xu, Bin Ouyang, Yuwei Li, Heye Zhang, Qi Song, Kunlin Cao, and Youbing Yin. Automated anatomical labeling of coronary arteries via bidirectional tree lstms. *International journal of computer assisted radiology and surgery*, 14(2):271–280, 2019.
- [23] Guanyu Yang, Alexander Broersen, Robert Petr, Pieter Kitslaar, Michiel A de Graaf, Jeroen J Bax, Johan HC Reiber, and Jouke Dijkstra. Automatic coronary artery tree labeling in coronary computed tomographic angiography datasets. In *2011 Computing in Cardiology*, pages 109–112. IEEE, 2011.
- [24] Han Yang, Junxuan Chen, Ying Chi, Xuansong Xie, and Xi-anheng Hua. Discriminative coronary artery tracking via 3d cnn in cardiac ct angiography. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 468–476. Springer, 2019.
- [25] Weidong Zhang, Jiamin Liu, Jianhua Yao, and Ronald M Summers. Automatic anatomical labeling of abdominal arteries for small bowel evaluation on 3d ct scans. In *2013 IEEE 10th international symposium on biomedical imaging*, pages 210–213. IEEE, 2013.
- [26] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.