Response letter

Dear Editor and Reviewers,

We are grateful for the comments and suggestions from the editors and the reviewers, which are crucial for improving our work. We have revised the manuscript to address the reviewers’ comments fully. Our point-by-point reply to the review comments is summarized below. In this document, the original reviewers’ comments are in **black**; our responses are in **blue**; the quotations in the revised manuscript are in **red**.

Reviewers' comments:

EiC: While you are revising your paper, here is a list of points worth checking, which we find author's overlook. I will check that these are adhered to before your paper is approved for publication, assuming the revision satisfies the Associate Editor and Reviewers.

Reply: Thanks. We have double-checked and revised the manuscript according to the list below from the editor-in-chief.

a) Make sure your title is succinct and grammatical. It should ideally not exceed 10-15 words.

Reply: Thanks for your suggestion; our current title, ‘AGMN: Association Graph-based Graph Matching Network for Coronary Artery Semantic Labeling on Invasive Coronary Angiograms,’ has 15 words precisely conveying the scope of our work.

b) Make sure your conclusions reflect on the strengths and weaknesses of your work, how others in the field can benefit from it and thoroughly discus future work. The conclusions should be different in content from the abstract and be rather longer too.

Reply: Thanks for the suggestion. Our current form of the conclusions follows the suggestions of the editor-in-chief. Specifically, we included a summary of our findings, discussed limitations, and outlined future work.

c) Take a careful look at your bibliography and how you cite papers listed in it. Make sure it is current and cites recent work. Please cite a variety of different sources of literature. Please do not make excessive citation to arXiv papers, or papers from a single conference series. Do not cite large groups of papers without individually commenting on them. So we discourage " In prior work [1,2,3,4,5,6] …". Your bibliography should only exceptionally exceed about 40 items.

Reply: Thanks for the suggestions. We have checked the reference section to meet the publication requirements. Besides, we have added several references accordingly and corrected the format of the citations.

d) You may have originally written your paper with a different audience in mind. Please make sure the revised version is relevant to the readership of Pattern Recognition. To this end, please make sure you cite RECENT work from the field of pattern recognition that will be relevant to our readership.

Reply: Thanks for the suggestion. Our study is related to deep learning on the graph and medical image processing for coronary arteries semantic labeling using invasive coronary angiograms, aiming at the potential readers in pattern recognition, especially for the special issue of ‘Graph Machine Learning.’

e) Do not exceed the page limits or violate the format, i.e. double spaced SINGLE column with a maximum of 35 pages for a regular paper and 40 pages for a review.

Reply: Our paper has a total page of 27 which meets the publication requirements.

GE:

1) Three reviewers pointed out some important suggestions and comments for the authors to consider. Please rigorously address all these comments and prepare a revision together with a one-to-one response letter.

Reply: Thanks for your reminder. This response letter elaborates on our point-by-point reply to the review comment. Also, we provide a revised manuscript with the highlighted changes and a clean version of the revised manuscript.

2) In Introduction, the main contribution and originality should be explained in more detail.

Reply: Thanks for your suggestions; we added two paragraphs to articulate our contributions and originality in the revised manuscript at the end of the introduction section.

(Section 1. Introduction)

To summarize, our work makes the following contributions:

1. This paper presents the first coronary arterial semantic labeling study on ICA images using graph matching, including both main arteries and side branches. Instead of performing pixel-to-pixel semantic segmentation, we proposed a pipeline with two major components: Arterial vascular tree pre-processing and a graph neural network named AGMN. Leveraging the semantic correspondence of the coronary arterial tree represented by a graph, the unlabeled arterial segments are classified by the labeled segments to achieve semantic labeling by graph matching.
2. The designed AGMN first performs the node and edge feature embeddings within the individual graphs and then extracts the cross-graph features using the assignment graph. Then a decoder module and a major voting layer are employed to perform vertex classification, a dual problem to coronary artery semantic labeling.
3. We demonstrate the robustness of the proposed coronary artery semantic labeling approach using the corrupted datasets.
4. We modify ZORRO [1], a graph neural network explainer, to explain the proposed AGMN for coronary artery semantic labeling by ranking the feature importance and node importance.

**Reviewer #1**: this paper describe a novel GNN based deep learning method on medical application. They evaluate the quality of GNN embedding via feature importance, which provide strong explainability to their method.

Thanks for your summary.

However, here are few potential improvements on this paper:

1. Since the classification has been done by DNN/GNN, how does the feature importance computed? Normally, feature importance are computed in tree-based method.

Reply: Thanks for your question. We agree with your illustration that feature importance is normally computed in conjunction with tree-based methods. In the previous version, the feature importance is computed by the leave-one-out technique [2], i.e., a feature is significant if the performance of semantic labeling decreases significantly when this feature is replaced by zero. The importance of the feature is obtained by removing a feature and then quantifying the changes in prediction accuracy. However, it can only explain one feature at a time and cannot draw an overall conclusion using a set of features.

We performed the literature review for the explanation techniques for graph neural networks. According to [3], the existing approaches for explaining graph neural networks (GNN) are categorized into 1) gradient-based approaches, 2) decomposition-based methods, 3) surrogate-based methods, and 4) perturbation-based methods.

1. Gradient-based approach. Class activation map (CAM) [4] is the most widely used approach for visualizing the attention RoIs and explaining the behavior for convolution neural network (CNN)-based networks, and it has been extended to explain GNN [5]. However, gradient-based methods suffer from saturation problems in that the model output changes minimally with respect to any input change. Thus, gradients may not represent an accurate reflection of the contributions of inputs [5]. In addition, CAM can only be applied to GNN model with a global average pooling (GAP) and a multi-layer perceptron (MLP) layer as the final classifier. Our AGMN only contains graph convolution network (GCN) layers, so the CAM method cannot be used to explain our graph-matching network.
2. Decomposition-based methods build score decomposition rules to distribute the prediction scores to the input space, such as GNN layer-wise relevance propagation (GNN-LRP) [6]. However, it can only explain the importance of nodes and cannot be applied to explain feature importance.
3. Surrogate-based methods employ a simple and interpretable surrogate model to approximate the predictions of the complex deep model for the neighboring areas of the input example [3]. They can only explain the importance of key nodes, rather than the importance of key features, such as GraphLime [7].
4. Perturbation-based methods compare the output variations with respect to different input perturbations to calculate the feature importance [3], such as ZORRO [1]. If the important features and nodes are retained, the prediction should be similar to the original one.

According to the above analysis, we employ one of the perturbation-based methods, ZORRO, to explain both the feature and node importance of our AGMN. ZORRO uses the discrete feature mask and node mask to recognize the important features and nodes in decision-making. Given an input graph, ZORRO iteratively and recursively adds important features and nodes according to the fidelity score, where the fidelity score measures the difference between the original predictions and the new predictions after masking out important features and important nodes [8].

We added the description of ZORRO in section 3.5 for feature importance interpretation.

(Section 3.5. Interpretability of AGMN)

We employ a perturbation-based method, ZORRO, to explain both the feature and node importance of our AGMN. ZORRO employs discrete feature masks and node masks to recognize the importance of features and nodes in decision-making. Given an input graph, ZORRO iteratively and recursively adds important features and nodes according to the fidelity score, where fidelity score measures the difference between the original predictions and the new predictions after masking out important features and important nodes [1,8].

ZORRO is a *post-hoc* algorithm that does not require post-training. Without the training process, a differentiable mask, i.e., a continuous mask, is no longer required. ZORRO employs the hard mask, where 0 indicates the feature or the node is not selected, and 1 indicates the feature or node is selected as the important input information. In addition, ZORRO explains the feature and node importance for each input graph at each time. As a hard mask selection algorithm, ZORRO aims at selecting the top-k frequent retrieved features and identifying the most important nodes for graph representation learning.

*Explaining feature importance*. ZORRO is proposed to explain the GNN for node classification task originally. Recall that we have converted the graph matching problem into a vertex classification task using the association graphs; identifying the concatenated features for each vertex cannot explain the feature importance for the node in the individual graph. Thus, we modify ZORRO to explain the feature importance of our AGMN. In our implementation, a unified feature mask is employed to mask out the selected features for and simultaneously. Then, the retained features from these two individual graphs are concatenated to form the features for each vertex in the association graph.

The results of calculated feature importance and the corresponding discussion were also added in the revised manuscript.

(Section 4.5. Feature importance)

In this study, we designed 121 hand-craft features, including pixel-intensity-based, positional, and topological features. We set the fidelity score threshold used in ZORRO as 0.8 to explain each pair of individual graphs used for graph matching. In our experiments, 45 and 40 subjects were selected as the testing set and template set, respectively. Since our AGMN requires that the two individual graphs are selected from the same view and the number of nodes is smaller or equal than , we obtained 700 association graphs during the testing. The frequency of the selected features is denoted as the feature importance. As shown in Figure 6, the frequency of 100% indicated the feature named *p2\_degree* was selected 700 times when explaining these 700 graph-matching pairs.

Chart

Description automatically generated

**Figure 6**. Feature importance ranking for classifying coronary arterial segments. Important features were surrogated by the frequency of the selected features during explaining all graph-matching pairs using the *post-hoc* GNN explaining algorithm, ZORRO [1]. The vertical axis indicates the feature names, while the horizontal axis indicates the frequencies of the features, which are also feature importance.

According to Figure 6, *p2\_degree* and *p1\_degree* were the most important features identified by ZORRO with a high frequency. The topological features, *p1\_degree* and *p2\_degree,* indicate the node degrees of the two endpoints of an artery segment. The *x\_center* and *y\_center* indicate the absolute centers of the segment positions to the center of the vascular tree. The *weighted\_x\_center* and *weighted\_y\_center* represent the weighted centers of the segment positions related to the center of the vascular tree. At the same time, *r\_min* and *r\_mean* indicate an arterial segment’s minimal and average radius, respectively. The detained definition of the hand-craft features is illustrated in Table S1.

Among these 15 features, 2 are topological features, one is pixel-intensity-based, and the other 12 are positional features. The results indicate that topological and positional features are the most important features for coronary artery semantic labeling in our AGMN. The results further strengthened our argument before, i.e., pixel-intensity-based models have difficulties distinguishing each arterial segment and generating semantic segmentation because of the morphological similarity among different branches in the coronary vascular tree and the overlap of the arteries in 2D. In contrast, our AGMN employed the positional and topological features and achieved high performance for coronary artery semantic labeling by comparing the features between different graphs.

We also reported the performance of AGMN using most frequently selected features, as shown in Figure 7. It was observed that using only the top 80 important features determined by ZORRO, our AGMN already achieved an ACC of 0.8089 compared to 0.8264 using all 121 features. With the limited number of features, such as one most-important feature (*p2\_degree*), our AGMN achieved an ACC of 0.5135 for classifying all types of arteries. For classifying LMA main branches, the proposed AGMN achieved an ACC of 0.7908 using the top 20 most important features. For LAD and LCX main branches, the AGMN achieved the PREC over 0.7 if using more than 20 most important features. The results indicated our AGMN achieved a satisfactory performance with limited features.

For classifying the side branches of D and OM, the AGMN achieved F1-scores over 0.7 when using the 80 most important features. However, if using 1 or 5 top important features, AGMN achieved the F1-scores below 0.5 for classifying side branches. These findings indicated that more features were required for AGMN to classify side branches than main branches.

Chart, line chart

Description automatically generated

**Figure 7**. Plots showing ACC, PREC, REC, and F1 of our proposed AGMN using different top-K features. We reported the average performance based on the five-fold cross-validation results. The performance for all types of arteries among different is annotated in blue dots and connected by the solid line; the performance for different types of arteries is annotated in asterisks with different colors and connected by dashed lines.

2. Some visualization of attention on different node might be helpful for understanding the model.

Reply: Thanks for your suggestion. The attention of different nodes is related to the node importance for node classification using GCN. We adopted the same algorithm used for feature importance explanation in section 4.5 in the revised manuscript, ZORRO [1], for node importance explanation. ZORRO iteratively and recursively adds important features and nodes according to the fidelity score. When adding a node to the node mask, ZORRO calculates the improvement of the fidelity score. We rank the node importance according to the improvement of fidelity scores each time post-adding a new node.

We also made several modifications to ZORRO to explain the node importance of our AGMN and added the modified ZORRO in the methodology section.

(Section 3.5. Interpretability of AGMN)

*Explaining node importance.* Identifying the vertex importance in the association graph cannot explain the importance of the nodes in the individual graphs. Thus, we attempted to use ZORRO to explain the importance of a node in the individual graphs. Specifically, all nodes in are retained, and all nodes in are removed at the initial stage. Then, we apply ZORRO to add a new node iteratively in to test the improvement of the fidelity score. Note that during the model testing, is constructed using ICA images from the testing set and is constructed using ICA images from the template set, as demonstrated in Algorithm 2. Iteratively adding nodes from the template graph explains the node importance for graph matching and coronary artery semantic labeling.

We also reported our results in Figure 5 and section 4.5 to discuss the node importance.

(Section 4.5. Feature importance)

We randomly selected three graph-matching pairs to visualize the results of our AGMN in Figure 5. We visualized the improvement in fidelity for each node for the final individual graph generated by switching nodes and edges in Figure 3 (f). The improvement of the fidelity scores is labeled in pseudo colors, ranging from 0 to 0.6 in our experiments. Among these cases, the LMA, the inlet of the coronary arterial tree, shows the highest node importance. The main branches, i.e., LAD1 and LCX1, which are directly connected to LMA, also show a higher fidelity score improvement than side branches, i.e., OM2 and D2. The results indicate that correctly identifying LMA is critical for labeling other branches. Our AGMN acts like interventional cardiologists who interpret coronary arteries from LMA main branch to the side branches.

**A picture containing text

Description automatically generated**

**Figure 5.** Graph matching results of three examples: (a) correspondence between coronary arteries from ICA images from the testing set (left) and from the template set (right). The green connection line indicates a correct match, and the red line represents a wrong match; (b) the corresponding node importance of the individual graph in the template set colored by the fidelity score improvement calculated by ZORRO.

**Reviewer #2:** In this paper, association graph-based graph matching network (AGMN) was proposed for coronary artery semantic labeling. In this network, coronary angiograms are used to extract vessel trees and construct association graphs, and graph convolution networks are used to learn similarities between arterial segments. Experiments on a corrupted dataset showed that the AGMN has good robustness.

Overall, the paper is relatively well written, but still have some problems with it as follows.

1. Figure 4 is more similar to Figure 2, and the details are not presented in enough detail. For example, how is G^A calculated in Ground Truth? What is the specific structure of module (c)? It is suggested to add the details of the proposed model in Figure 4.

Thanks for your questions and suggestions. We reorganized Figure 4 in the revised manuscript, and now Figure 4 illustrates the graph-matching process using our AGMN. The structure of module (c) is depicted in detailed architecture in the revised version. Also, we added the description of ground truth generation to the supplemental material. And an example of generating the ground truth generation is shown in section 2 in supplemental materials in the revised version.

(Section 3.2. Graph matching neural network for arterial segment labeling)

A screenshot of a computer

Description automatically generated with low confidence

**Figure 4.** A flowchart illustrating the architecture of the association graph-based graph matching network (AGMN): **(a)** individual graph generation, node feature extraction, and edge feature extraction. Each node is colored, and its features are shown in rectangles. **(b)** association graph generation, ground truth generation, and feature concatenation for vertices in the association graph. The node correspondence is generated by the nodes’ labels from the individual graphs in (a). The positive vertices are annotated in green, whereas negative vertices are annotated in yellow. **(c)** AGMN. The AGMN includes a feature embedding module implemented by multi-layer perception (MLP), a graph convolution network, and a feature decoding module implemented by MLPs. The node features in (a) are used to train the AGMN. The vertex classification is achieved by majority voting. The blue rectangle in (c) represents MLP layers.

An example of graph matching is shown in Figure 4. In Figure 4 (a), the individual graphs (with 3 nodes and 2 edges) and (with 3 nodes and 3 edges) are used to construct the association graph with 9 vertices and 12 edges. According to the semantic labels of the nodes in the individual graphs, which are represented by different colors, the vertex , and are labeled as positive vertices (in green), while the other vertices are labeled as negative vertices (in yellow). The vertex classification results are represented as the permutation matrix . The permutation loss computed by the cross entropy between the predicted vertex class and the ground truth is used as an objective function, as shown in Eq. 9.

The process of generating ground truth has been included in the supplemental materials.

(Supplemental materials: Section 2. Ground truth generation for graph matching)

For the ground truth generation, we adopted two rules:

1. We first manually annotated the ICA images with semantic labels for each arterial segment. Then, the semantic label was assigned to each node (i.e., each arterial segment) in the individual graph. During the creation of the database, the node correspondences between arterial segments are automatically identified based on the semantic labels, i.e. if two arterial segments have the same types.
2. However, the main branches, such as LCX and LAD, are separated into several small branches during the individual graph generation. Then, the arterial branches with the same semantic labels may have more than one node in the individual graph. For example, in Figure 3 (d), the LAD branch contains two segments due to the bifurcation points connected to side branch D1. These segments are matched with the LAD segments from another individual graph, i.e. .

An example of generating the ground truth generation is shown in Figure S1. In Figure S1, we provided two ICA images with semantic labels for each arterial branch. The semantic label was assigned to each node (i.e., each arterial segment) in the individual graph during the manual annotation process. According to the semantic label, the ground truth was generated according to the node correspondence. In this example, contains arterial branches, and contains arterial branches; thus, an 2D array with the dimension of is used as the ground truth for AGMN, where ‘1’ indicates the matched nodes and ‘0’ indicates the unmatched nodes.

A picture containing text

Description automatically generated

Figure S1. An illustrative example showing how the ground truth is generated for training AGMN.

2. How are G\_1,G\_2 generated in the methodology section?

Reply: Thanks for your question. In Figure 2 (a), we depict the proposed coronary artery preprocessing workflow. Coronary arterial contours are extracted using our previously developed Feature Pyramid U-Net++ (FP-U-Net++) [9]. Then, each centerline is extracted to reduce redundant foreground pixels in a binary image while preserving the connectivity and topology of the vascular tree. As a result, the vascular tree's centerline and the arterial segments' diameters are calculated. We also design five rules and an algorithm to eliminate errors and build the individual graph, including capillary segments deletion, splitting point merging, cycle deletion, degree two points merging, and node and edge switch.And an example is provided in Figure 3, where the Figure 3 (f) shows the final individual graph, i.e., and in our algorithm.

A picture containing polygon

Description automatically generated

**Figure 3.** Workflow of individual graph generation and vascular centerline post-processing: **(a)** original ICA image; **(b)** centerline extraction and key point detection. The bifurcation points with degree > 3 are marked in red, and the endpoints with degree = 1 are marked in green; **(c)** vascular structure after merging the splitting points in the yellow region in (b) and deleting the cycle in the blue region in (b). In this sub-figure, the bifurcation points with degree > 3 are marked in red, the endpoints with degree = 1 are marked in green, and the degree two points generated by merging the splitting points are marked in blue; **(d)** vascular structure after merging the degree two points in (c); **(e)** generated individual graph; **(f)** the final individual graph by switching nodes and edges of the individual graph in (e). Since the graph matching is to find the node (arterial segment) correspondence rather than the edge correspondence, we switch node and edge in the individual graph in (e). In (c), (d), and (f), the classes of the arterial segments are labeled; in (c), (d), and (e), the node indices are annotated.

3. In the methodology section, the focus is not clear enough, and the innovation point of the paper is not introduced clearly.

Thanks for pointing out the issues. We add a short paragraph at the beginning of the methodology section to strengthen the focal points, as follows.

The approach presented in this study focuses on segment-level classification for coronary artery semantic labeling. We first design a workflow to preprocess the ICA image and generate the individual graph for further use; then, we develop an association graph-based graph-matching neural network to learn the similarities between arterial segments from two (arterial) individual graphs. Using AGMN, the problem of the semantic labeling task is converted into an equivalent problem of finding one-to-one or one-to-zero mapping for arterial segments from two different individual graphs. We also develop training and testing algorithms to label coronary artery semantics using the proposed AGMN. Figure 1 illustrates the overall workflow and details in subsequent sections below.

(Section 1. Introduction)

To summarize, our work makes the following contributions:

1. This paper presents the first coronary arterial semantic labeling study on ICA images using graph matching, including both main arteries and side branches. Instead of performing pixel-to-pixel semantic segmentation, we proposed a pipeline with two major components: Arterial vascular tree pre-processing and a graph neural network named AGMN. Leveraging the semantic correspondence of the coronary arterial tree represented by a graph, the unlabeled arterial segments are classified by the labeled segments to achieve semantic labeling by graph matching.
2. The designed AGMN first performs the node and edge feature embeddings within the individual graphs and then extracts the cross-graph features using the assignment graph. Then a decoder module and a major voting layer are employed to perform vertex classification, a dual problem to coronary artery semantic labeling.
3. We demonstrate the robustness of the proposed coronary artery semantic labeling approach using the corrupted datasets.
4. We modify ZORRO [1], a graph neural network explainer, to explain the proposed AGMN for coronary artery semantic labeling by ranking the feature importance and node importance.

4. In the introduction, the challenge of semantic labeling of coronary angiograms is not described clearly enough.

Reply: Thanks for pointing out the issue. In the revised manuscript, we added the demonstration of the challenge of semantic labeling of coronary arteries in the introduction section.

(Section 1. Introduction)

It is significant to note that semantic segmentation of ICA images is more complex for the following practical reasons: 1) contrast degradation due to dissipation of contrast dye, image noise, limited radiation dosage, or X-ray exposure; 2) spatial blurring due to X-ray imaging devices, and temporal blurring due to system lag in real-time imaging; and 3) object overlaps due to the very limited view angles.

Pixel-intensity-based models present difficulties distinguishing between each arterial segment and generating semantic segmentation given the morphological similarity among branches in the coronary vascular tree and arterial overlap in 2D (projection) ICA, as shown in Figure 1 (b). Also, the coronary arteries not only span a long distance but also share similar semantic features [10], making it challenging to associate them with their exact branches. False identifications of arterial segments not only impair the understanding of the structure of the arterial tree but also may lead to inaccurate assessment of disease description in the clinical workflow [11]. Existing methodologies that rely solely on position and imaging features may produce unsatisfactory results when processing complicated coronary vasculature [12]. The topology is a crucial factor in arterial identification, inspiring us to convert arteries and their connectivity into graphs and perform coronary artery semantic labeling using graphs.

5. At the end of the introduction, the organization of the whole paper is not introduced.

Reply: Thanks for your reminder. We added the organization of the manuscript below.

(Section 1. Introduction)

This paper is organized as follows. Section 1 introduces the background and challenges of coronary artery semantic labeling. Section 2 reviews existing algorithms on coronary artery semantic segmentation and graph matching in recent years. In Section 3, the proposed AGMN is described in detail. The enrolled subjects, implementation details, experimental results, and discussion are presented in Section 4. Section 5 illustrates the limitations and future work, followed by some closing marks in Section 6.

6. In Figure 6, two features, p2\_degree and p1\_degree, have a significant impact on the results. How to explain this result? Is it possible to use only the more influential features in the model?

Reply: Thanks for your question. We agree with your illustration that feature importance is normally computed in conjunction with tree-based methods. In the previous version, the feature importance is computed by the leave-one-out technique [2], i.e., a feature is significant if the performance of semantic labeling decreases significantly when this feature is replaced by zero. The importance of the feature is obtained by removing a feature and then quantifying the changes in prediction accuracy. However, it can only explain one feature at a time and cannot draw an overall conclusion using a set of features.

We performed the literature review for the explanation techniques for graph neural networks. According to [3], the existing approaches for explaining graph neural networks (GNN) are categorized into 1) gradient-based approaches, 2) decomposition-based methods, 3) surrogate-based methods, and 4) perturbation-based methods.

1. Gradient-based approach. Class activation map (CAM) [4] is the most widely used approach for visualizing the attention RoIs and explaining the behavior for convolution neural network (CNN)-based networks, and it has been extended to explain GNN [5]. However, gradient-based methods suffer from saturation problems in that the model output changes minimally with respect to any input change. Thus, gradients may not represent an accurate reflection of the contributions of inputs [5]. In addition, CAM can only be applied to GNN model with a global average pooling (GAP) and a multi-layer perceptron (MLP) layer as the final classifier. Our AGMN only contains graph convolution network (GCN) layers, so the CAM method cannot be used to explain our graph-matching network.
2. Decomposition-based methods build score decomposition rules to distribute the prediction scores to the input space, such as GNN layer-wise relevance propagation (GNN-LRP) [6]. However, it can only explain the importance of nodes and cannot be applied to explain feature importance.
3. Surrogate-based methods employ a simple and interpretable surrogate model to approximate the predictions of the complex deep model for the neighboring areas of the input example [3]. They can only explain the importance of key nodes, rather than the importance of key features, such as GraphLime [7].
4. Perturbation-based methods compare the output variations with respect to different input perturbations to calculate the feature importance [3], such as ZORRO [1]. If the important features and nodes are retained, the prediction should be similar to the original one.

According to the above analysis, we employ one of the perturbation-based methods, ZORRO, to explain the feature importance of our AGMN. ZORRO employs the discrete feature mask and node mask to recognize the important features and nodes in decision-making. Given an input graph, ZORRO iteratively and recursively adds important features and nodes according to the fidelity score, where the fidelity score measures the difference between the original predictions and the new predictions after masking out important features and important nodes [8].

We added the description of ZORRO in section 3.5 for feature importance interpretation.

(Section 3.5. Interpretability of AGMN)

We employ a perturbation-based method, ZORRO , to explain both the feature and node importance of our AGMN. ZORRO employs discrete feature masks and node masks to recognize the importance of features and nodes in decision-making. Given an input graph, ZORRO iteratively and recursively adds important features and nodes according to the fidelity score, where fidelity score measures the difference between the original predictions and the new predictions after masking out important features and important nodes [1,8].

ZORRO is a *post-hoc* algorithm that does not require post-training. Without the training process, a differentiable mask, i.e., a continuous mask, is no longer required. ZORRO employs the hard mask, where 0 indicates the feature or the node is not selected, and 1 indicates the feature or node is selected as the important input information. In addition, ZORRO explains the feature and node importance for each input graph at each time. As a hard mask selection algorithm, ZORRO aims at selecting the top-k frequent retrieved features and identifying the most important nodes for graph representation learning.

*Explaining feature importance*. ZORRO is proposed to explain the GNN for node classification task originally. Recall that we have converted the graph matching problem into a vertex classification task using the association graphs; identifying the concatenated features for each vertex cannot explain the feature importance for the node in the individual graph. Thus, we modify ZORRO to explain the feature importance of our AGMN. In our implementation, a unified feature mask is employed to mask out the selected features for and simultaneously. Then, the retained features from these two individual graphs are concatenated to form the features for each vertex in the association graph.

The results of calculated feature importance and the corresponding discussion were also added in the revised manuscript.

(Section 4.5. Feature importance)

In this study, we designed 121 hand-craft features, including pixel-intensity-based, positional, and topological features. We set the fidelity score threshold used in ZORRO as 0.8 to explain each pair of individual graphs used for graph matching. In our experiments, 45 and 40 subjects were selected as the testing set and template set, respectively. Since our AGMN requires that the two individual graphs are selected from the same view and the number of nodes is smaller or equal than , we obtained 700 association graphs during the testing. The frequency of the selected features is denoted as the feature importance. As shown in Figure 6, the frequency of 100% indicated the feature named *p2\_degree* was selected 700 times when explaining these 700 graph-matching pairs.

Chart

Description automatically generated

**Figure 6**. Feature importance ranking for classifying coronary arterial segments. Important features were surrogated by the frequency of the selected features during explaining all graph-matching pairs using the *post-hoc* GNN explaining algorithm, ZORRO [1]. The vertical axis indicates the feature names, while the horizontal axis indicates the frequencies of the features, which are also feature importance.

According to Figure 6, *p2\_degree* and *p1\_degree* were the most important features identified by ZORRO with a high frequency. The topological features, *p1\_degree* and *p2\_degree,* indicate the node degrees of the two endpoints of an artery segment. The *x\_center* and *y\_center* indicate the absolute centers of the segment positions to the center of the vascular tree. The *weighted\_x\_center* and *weighted\_y\_center* represent the weighted centers of the segment positions related to the center of the vascular tree. At the same time, *r\_min* and *r\_mean* indicate an arterial segment’s minimal and average radius, respectively. The detained definition of the hand-craft features is illustrated in Table S1.

Among these 15 features, 2 are topological features, one is pixel-intensity-based, and the other 12 are positional features. The results indicate that topological and positional features are the most important features for coronary artery semantic labeling in our AGMN. The results further strengthened our argument before, i.e., pixel-intensity-based models have difficulties distinguishing each arterial segment and generating semantic segmentation because of the morphological similarity among different branches in the coronary vascular tree and the overlap of the arteries in 2D. In contrast, our AGMN employed the positional and topological features and achieved high performance for coronary artery semantic labeling by comparing the features between different graphs.

We also reported the performance of AGMN using most frequently selected features, as shown in Figure 7. It was observed that using only the top 80 important features determined by ZORRO, our AGMN already achieved an ACC of 0.8089 compared to 0.8264 using all 121 features. With the limited number of features, such as one most-important feature (*p2\_degree*), our AGMN achieved an ACC of 0.5135 for classifying all types of arteries. For classifying LMA main branches, the proposed AGMN achieved an ACC of 0.7908 using the top 20 most important features. For LAD and LCX main branches, the AGMN achieved the PREC over 0.7 if using more than 20 most important features. The results indicated our AGMN achieved a satisfactory performance with limited features.

For classifying the side branches of D and OM, the AGMN achieved F1-scores over 0.7 when using the 80 most important features. However, if using 1 or 5 top important features, AGMN achieved the F1-scores below 0.5 for classifying side branches. These findings indicated that more features were required for AGMN to classify side branches than main branches.

Chart, line chart

Description automatically generated

**Figure 7**. Plots showing ACC, PREC, REC, and F1 of our proposed AGMN using different top-K features. We reported the average performance based on the five-fold cross-validation results. The performance for all types of arteries among different is annotated in blue dots and connected by the solid line; the performance for different types of arteries is annotated in asterisks with different colors and connected by dashed lines.

Also, we added the detailed explanation of these 121 hand-crafted feature in the supplemental materials (Supplemental materials, Section 1. Detailed explanation for the handcraft features for each arterial segment)

7. The images in the text are of different sizes, and a reasonable layout is suggested. For example, Figure 3 and Figure 7 take up too much space.

Thanks for pointing out the issues. We changed the text font for Figure 1, Figure 2, Figure 4. And we also rearranged the subgraphs in Figure 3 and Figure 7. Figure 3 in the revised version is shown below.

Chart

Description automatically generated

**Figure 3.** Workflow of individual graph generation and vascular centerline post-processing: **(a)** original ICA image; **(b)** centerline extraction and key point detection. The bifurcation points with degree > 3 are marked in red, and the endpoints with degree = 1 are marked in green; **(c)** vascular structure after merging the splitting points in the yellow region in (b) and deleting the cycle in the blue region in (b). In this sub-figure, the bifurcation points with degree > 3 are marked in red, the endpoints with degree = 1 are marked in green, and the degree two points generated by merging the splitting points are marked in blue; **(d)** vascular structure after merging the degree two points in (c); **(e)** generated individual graph; **(f)** the final individual graph by switching nodes and edges of the individual graph in (e). Since the graph matching is to find the node (arterial segment) correspondence rather than the edge correspondence, we switch node and edge in the individual graph in (e). In (c), (d), and (f), the classes of the arterial segments are labeled; in (c), (d), and (e), the node indices are annotated.

Figure 7 in the revised manuscript is shown below.

Chart, line chart

Description automatically generated

**Figure 8.** The achieved ACC, PREC, REC, and F1 of the proposed AGMN, CPR-GCN, and BiTreeLSTM using different corrupted datasets. The horizontal axis indicates the probability of deleting an artery segment randomly.

8. the table layout of the text is rough, suggesting the use of three-line tables.

Thanks for your suggestion. We use three-line tables in the revised version for each table. Table 3 now reads

**Table 3**. Hyperparameter settings in the grid search

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Search space | Description |
| Number of hidden units in MLP | [16, 32, 64] | The MLP includes the feature embedding module, GCN, and feature decoder module. For each experiment, these two hyperparameters are identically set to all MLP layers. |
| Number of MLP layers | [2, 3, 4] |
| Number of the message passing steps () | [2, 3, 4] | The number of the message passing steps indicates the update iterations of the GCN module in GMN. |
| Number of samples in the template set () | [27, 40, 52, 79] | 10%, 15%, 20% and 30% of the ICA images were selected using the stratified sampling according to the view angles as the template set. |

Table 4 now reads

**Table 4.** A summary of the best performance achieved by our proposed AGMN for coronary artery semantic labeling. LMA, left main artery; LAD, left descending artery; LCX, left circumflex artery; D, diagonal artery; OM obtuse margin.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artery type | ACC | PRE | REC | F1 |
| LMA | 0.9956±0.0089 | 0.9911±0.0109 | 0.9956±0.0089 | 0.9933±0.0089 |
| LCX | 0.8432±0.0306 | 0.8476±0.0481 | 0.8432±0.0306 | 0.8452±0.0386 |
| LAD | 0.8046±0.0452 | 0.8256±0.0307 | 0.8046±0.0452 | 0.8143±0.0310 |
| D | 0.7956±0.0412 | 0.7536±0.0493 | 0.7956±0.0412 | 0.7736±0.0424 |
| OM | 0.7565±0.0825 | 0.7613±0.0319 | 0.7565±0.0825 | 0.7569±0.0508 |
| All | 0.8264±0.0302 | 0.8276±0.0298 | 0.8264±0.0302 | 0.8262±0.0301 |

Table 5 now reads

**Table 5.** Comparisons between baseline methods and our proposed AGMN for coronary artery semantic labeling using our ICA dataset. The means and standard deviations of the accuracy, precision, recall, and F1-scores among the five folds are presented. The bold texts indicate they achieved the best performance in their corresponding evaluation metrics.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metric | Method | LMA | LAD | LCX | D | OM | All |
| ACC | SVM | 0.9925±0.0151 | 0.6331±0.0540 | 0.6388±0.0390 | 0.6147±0.0410 | 0.5907±0.0419 | 0.6651±0.0080 |
| UTD | 1.0000±0.0000 | 0.8828±0.0136 | 0.9245±0.0235 | 0.0000±0.0000 | 0.0000±0.0000 | 0.6182±0.0094 |
| DTU | 0.0000±0.0000 | **1.0000±0.0000** | 0.7359±0.0194 | 0.0000±0.0000 | 0.3575±0.2923 | 0.5450±0.0516 |
| BiTreeLSTM | 1.0000±0.0000 | 0.8845±0.0150 | 0.9871±0.0120 | 0.0000±0.0000 | 0.5981±0.0165 | 0.7492±0.0085 |
| CPR-GCN | 0.5361±0.2996 | 0.5319±0.1239 | 0.5072±0.1447 | 0.0624±0.0953 | 0.5341±0.3045 | 0.4581±0.0536 |
| AGMN | **0.9956±0.0089** | 0.8432±0.0306 | **0.8046±0.0452** | **0.7956±0.0412** | **0.7565±0.0825** | **0.8264±0.0302** |
| PRE | SVM | 0.9778±0.0071 | 0.6586±0.0174 | 0.6375±0.0378 | 0.5554±0.0101 | 0.6278±0.0261 | 0.6679±0.0081 |
| UTD | 1.0000±0.0000 | 0.7927±0.0167 | 0.4629±0.0106 | 0.0000±0.0000 | 0.0000±0.0000 | 0.4562±0.0069 |
| DTU | 1.0000±0.0000 | **0.8562±0.0190** | 0.5853±0.0099 | 0.0000±0.0000 | 0.9808±0.0122 | 0.6927±0.0074 |
| BiTreeLSTM | 0.0000±0.0000 | 0.4284±0.0044 | 0.6795±0.1010 | 0.0000±0.0000 | 0.5924±0.4837 | 0.4264±0.1212 |
| CPR-GCN | 0.6208±0.3240 | 0.5675±0.0540 | 0.3964±0.0570 | 0.2802±0.3727 | 0.3821±0.0139 | 0.4463±0.1075 |
| AGMN | **0.9911±0.0109** | 0.8476±0.0481 | **0.8256±0.0307** | **0.7536±0.0493** | **0.7613±0.0319** | **0.8276±0.0298** |
| REC | SVM | 0.9925±0.0151 | 0.6331±0.0540 | 0.6388±0.0390 | 0.6147±0.0410 | 0.5907±0.0419 | 0.6651±0.0080 |
| UTD | 1.0000±0.0000 | 0.8828±0.0136 | 0.9245±0.0235 | 0.0000±0.0000 | 0.0000±0.0000 | 0.6182±0.0094 |
| DTU | 0.0000±0.0000 | **1.0000±0.0000** | 0.7359±0.0194 | 0.0000±0.0000 | 0.3575±0.2923 | 0.5450±0.0516 |
| BiTreeLSTM | 1.0000±0.0000 | 0.8845±0.0150 | **0.9871±0.0120** | 0.0000±0.0000 | 0.5981±0.0165 | 0.7492±0.0085 |
| CPR-GCN | 0.5361±0.2996 | 0.5319±0.1239 | 0.5072±0.1447 | 0.0624±0.0953 | 0.5341±0.3045 | 0.4581±0.0536 |
| AGMN | **0.9956±0.0089** | 0.8432±0.0306 | 0.8046±0.0452 | **0.7956±0.0412** | **0.7565±0.0825** | **0.8264±0.0302** |
| F1 | SVM | 0.9850±0.0076 | 0.6437±0.0213 | 0.6360±0.0087 | 0.5832±0.0234 | 0.6071±0.0183 | 0.6646±0.0077 |
| UTD | 1.0000±0.0000 | 0.8353±0.0142 | 0.6169±0.0143 | 0.0000±0.0000 | 0.0000±0.0000 | 0.5135±0.0075 |
| DTU | 0.0000±0.0000 | 0.5998±0.0043 | 0.7018±0.0532 | 0.0000±0.0000 | 0.4458±0.3643 | 0.4491±0.0838 |
| BiTreeLSTM | 1.0000±0.0000 | **0.8699±0.0101** | 0.7348±0.0093 | 0.0000±0.0000 | 0.7429±0.0141 | 0.6967±0.0085 |
| CPR-GCN | 0.5698±0.3026 | 0.5455±0.0899 | 0.4353±0.0632 | 0.0742±0.0957 | 0.3924±0.1660 | 0.4192±0.0661 |
| AGMN | **0.9933±0.0089** | 0.8452±0.0386 | **0.8143±0.0310** | **0.7736±0.0424** | **0.7569±0.0508** | **0.8262±0.0301** |

9. The format of references is not uniform, such as [10][11][13][20][21][25][26][27][35][37].

Thanks for your reminder. We have checked the format of the references and changed them to the style of Pattern Recognition Journal. According to the instruction (<https://www.elsevier.com/journals/pattern-recognition/0031-3203/guide-for-authors>), we listed the author names, the paper title and the abbreviation of the journal names and related volume, pages and year; for the conference paper, we added the full name of the conference, place and years.

**Reviewer #4:** In this manuscript, the authors have proposed an association graph-based graph matching network (AGMN) for coronary arterial semantic labeling. Experimental results demonstrate that the proposed AGMN model achieved an average accuracy of 0.8264, an average precision of 0.8276, an average recall of 0.8264, and an average F1-score of 0.8262, which significantly outperformed existing coronary artery semantic labeling methods. The research topic is very interesting and the whole manuscript is well-organized. However, there are some minor comments:

1. For Table 4, it is better to put the same metric together for better comparison and readability.

Thanks for your suggestions. In the revised version, we rearranged this table by combining the same metric for better comparison and readability. Readers can compare one metric clearly without jumping between different rows. Table 5 now reads

**Table 5.** Comparisons between baseline methods and our proposed AGMN for coronary artery semantic labeling using our ICA dataset. The means and standard deviations of the accuracy, precision, recall, and F1-scores among the five folds are presented. The bold texts indicate they achieved the best performance in their corresponding evaluation metrics.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metric | Method | LMA | LAD | LCX | D | OM | All |
| ACC | SVM | 0.9925±0.0151 | 0.6331±0.0540 | 0.6388±0.0390 | 0.6147±0.0410 | 0.5907±0.0419 | 0.6651±0.0080 |
| UTD | 1.0000±0.0000 | 0.8828±0.0136 | 0.9245±0.0235 | 0.0000±0.0000 | 0.0000±0.0000 | 0.6182±0.0094 |
| DTU | 0.0000±0.0000 | **1.0000±0.0000** | 0.7359±0.0194 | 0.0000±0.0000 | 0.3575±0.2923 | 0.5450±0.0516 |
| BiTreeLSTM | 1.0000±0.0000 | 0.8845±0.0150 | 0.9871±0.0120 | 0.0000±0.0000 | 0.5981±0.0165 | 0.7492±0.0085 |
| CPR-GCN | 0.5361±0.2996 | 0.5319±0.1239 | 0.5072±0.1447 | 0.0624±0.0953 | 0.5341±0.3045 | 0.4581±0.0536 |
| AGMN | **0.9956±0.0089** | 0.8432±0.0306 | **0.8046±0.0452** | **0.7956±0.0412** | **0.7565±0.0825** | **0.8264±0.0302** |
| PRE | SVM | 0.9778±0.0071 | 0.6586±0.0174 | 0.6375±0.0378 | 0.5554±0.0101 | 0.6278±0.0261 | 0.6679±0.0081 |
| UTD | 1.0000±0.0000 | 0.7927±0.0167 | 0.4629±0.0106 | 0.0000±0.0000 | 0.0000±0.0000 | 0.4562±0.0069 |
| DTU | 1.0000±0.0000 | **0.8562±0.0190** | 0.5853±0.0099 | 0.0000±0.0000 | 0.9808±0.0122 | 0.6927±0.0074 |
| BiTreeLSTM | 0.0000±0.0000 | 0.4284±0.0044 | 0.6795±0.1010 | 0.0000±0.0000 | 0.5924±0.4837 | 0.4264±0.1212 |
| CPR-GCN | 0.6208±0.3240 | 0.5675±0.0540 | 0.3964±0.0570 | 0.2802±0.3727 | 0.3821±0.0139 | 0.4463±0.1075 |
| AGMN | **0.9911±0.0109** | 0.8476±0.0481 | **0.8256±0.0307** | **0.7536±0.0493** | **0.7613±0.0319** | **0.8276±0.0298** |
| REC | SVM | 0.9925±0.0151 | 0.6331±0.0540 | 0.6388±0.0390 | 0.6147±0.0410 | 0.5907±0.0419 | 0.6651±0.0080 |
| UTD | 1.0000±0.0000 | 0.8828±0.0136 | 0.9245±0.0235 | 0.0000±0.0000 | 0.0000±0.0000 | 0.6182±0.0094 |
| DTU | 0.0000±0.0000 | **1.0000±0.0000** | 0.7359±0.0194 | 0.0000±0.0000 | 0.3575±0.2923 | 0.5450±0.0516 |
| BiTreeLSTM | 1.0000±0.0000 | 0.8845±0.0150 | **0.9871±0.0120** | 0.0000±0.0000 | 0.5981±0.0165 | 0.7492±0.0085 |
| CPR-GCN | 0.5361±0.2996 | 0.5319±0.1239 | 0.5072±0.1447 | 0.0624±0.0953 | 0.5341±0.3045 | 0.4581±0.0536 |
| AGMN | **0.9956±0.0089** | 0.8432±0.0306 | 0.8046±0.0452 | **0.7956±0.0412** | **0.7565±0.0825** | **0.8264±0.0302** |
| F1 | SVM | 0.9850±0.0076 | 0.6437±0.0213 | 0.6360±0.0087 | 0.5832±0.0234 | 0.6071±0.0183 | 0.6646±0.0077 |
| UTD | 1.0000±0.0000 | 0.8353±0.0142 | 0.6169±0.0143 | 0.0000±0.0000 | 0.0000±0.0000 | 0.5135±0.0075 |
| DTU | 0.0000±0.0000 | 0.5998±0.0043 | 0.7018±0.0532 | 0.0000±0.0000 | 0.4458±0.3643 | 0.4491±0.0838 |
| BiTreeLSTM | 1.0000±0.0000 | **0.8699±0.0101** | 0.7348±0.0093 | 0.0000±0.0000 | 0.7429±0.0141 | 0.6967±0.0085 |
| CPR-GCN | 0.5698±0.3026 | 0.5455±0.0899 | 0.4353±0.0632 | 0.0742±0.0957 | 0.3924±0.1660 | 0.4192±0.0661 |
| AGMN | **0.9933±0.0089** | 0.8452±0.0386 | **0.8143±0.0310** | **0.7736±0.0424** | **0.7569±0.0508** | **0.8262±0.0301** |

2. It is better to further discuss the computational complexity of the proposed method with other existing methods.

Thanks for your suggestion. We added a new section (4.7 Computational complexity) to the revised manuscript. In the revised manuscript, we provide additional information: the number of parameters, training times, and inference times of the proposed AGMN and other existing methods. We also added a new section in the supplemental materials to demonstrate the implementation details of the baseline models.

As illustrated in section 3.3, ‘Each individual graph from the test set is paired with the individual graph from one representative subject in the template set for graph matching. In the template set, each arterial segment is labeled for reference. Using the well-trained AGMN, the mapping relationship between unlabeled arteries in the test subject and the labeled arteries from the template subject is obtained. The vertex classification result for the test subject among the subjects in the template set is voted based on maximum voting.’, our AGMN has the longest inference time. In algorithm 2, for the testing, AGMN needs to compare the testing graph with all graphs from the template sets, which results in a long inference time. This is a limitation of our AGMN, as illustrated in section 5, ‘The limitation of the proposed AGMN is that during the prediction, the matching graph procedures are required to be performed between the test subject and every subject in the template set.’ It is our future work that ‘graph clustering will be used to select the most representative subjects in each cluster and then construct the template set to accelerate the prediction’.

(Section 4.7. Computational complexity)

To evaluate the model complexity, we compared the number of weights for each model in Table 6, except for the machine learning-based model. For the BiTreeLSTM, UTD, and DTU models, we set the hidden layer with a size of 128 and the LSTM layer with a hidden size of 30. Under this setting, the number of weights was only 18.3K to 25.8K. The CPR-GCN contains the MLP layers for positional feature extraction and CNN and LSTM layers for imaging feature extraction. In our implemented CPR-GCN, three MLP layers were used for positional feature extraction, a convolutional bi-directional LSTM (CBiLSTM) was employed for imaging feature extraction, and a GNN with two GCN layers was used for feature aggregation. The number of weights for CPR-GCN was 2.46M, which was 7 times larger than that of AGMN. Our AGMN only contains MLP and GCN layers, so limited weights are required. Though with a limited number of weights compared to CPR-GCN, our model achieved the highest coronary artery semantic labeling performance. The implementation details of the baseline models are provided in supplementary materials.

We also reported the training time and inference time consumption in Table 6. The machine learning-based method only employed a support vector machine (SVM), which can be trained in less than one minute. The tree LSTM-based methods, such as UTD, DTU, and BiTreeLSTM, only employed one MLP layer to perform feature embedding for each arterial segment and a tree-structured LSTM for sequential feature extraction. In our experiments, we set the hidden size of these three models as 128. With limited model sizes, the UTD and DTU were trained in 0.25 hours, and the BiTreeLSTM was trained in 0.5 hours. All models converged within 200 epochs. For the CPR-GCN, with the use of cropped image patches and CNNs, the number of weights was significantly increased, resulting in a longer training time. Since these baseline methods were not publicly available, we added the implementation details in the supplemental materials. As illustrated in section 3.3, for the testing, AGMN needs to compare the testing graph with all graphs from the template sets, which results in a long inference time.

**Table 6.** Comparison of the computation complexity, training time and inference time between our AGMN and existing models for coronary artery semantic labeling using ICAs

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Number of weights | Training time (hours) | Inference time (seconds) |
| SVM | - | 0.01 | 0.001 |
| UTD | 18.30K | 0.25 | 0.008 |
| DTU | 18.30K | 0.25 | 0.008 |
| BiTreeLSTM | 25.80K | 0.50 | 0.016 |
| CPR-GCN | 2.46M | 9.60 | 0.415 |
| Our AGMN | 0.30M | 6.10 | 0.613 |

**Reference for response letter**

[1] T. Funke, M. Khosla, M. Rathee, A. Anand, ZORRO: Valid, Sparse, and Stable Explanations in Graph Neural Networks, IEEE Trans. Knowl. Data Eng. (2023) 1–12. https://doi.org/10.1109/TKDE.2022.3201170.

[2] T. Wang, W. Shao, Z. Huang, H. Tang, J. Zhang, Z. Ding, K. Huang, MOGONET integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification, Nat. Commun. 12 (2021) 3445. https://doi.org/10.1038/s41467-021-23774-w.

[3] H. Yuan, H. Yu, S. Gui, S. Ji, Explainability in Graph Neural Networks: A Taxonomic Survey, IEEE Trans. Pattern Anal. Mach. Intell. (2022) 1–19. https://doi.org/10.1109/TPAMI.2022.3204236.

[4] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: 2016 IEEE Conf Computer Vision and Pattern Recognitition, Las Vegas, NV, 2016: pp. 2921–2929.

[5] P.E. Pope, S. Kolouri, M. Rostami, C.E. Martin, H. Hoffmann, Explainability Methods for Graph Convolutional Neural Networks, in: 2016 IEEE Conf Computer Vision and Pattern Recognitition, Long Beach, CA, 2019: pp. 10772–10781.

[6] T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K.T. Schütt, K.-R. Müller, G. Montavon, Higher-order explanations of graph neural networks via relevant walks, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2021) 7581–7596.

[7] Q. Huang, M. Yamada, Y. Tian, D. Singh, Y. Chang, Graphlime: Local interpretable model explanations for graph neural networks, IEEE Trans. Knowl. Data Eng. (2022).

[8] S. Hooker, D. Erhan, P.-J. Kindermans, B. Kim, A Benchmark for Interpretability Methods in Deep Neural Networks, in: 2019 International Conference on Neural Information Processing Systems, (NIPS), Vancouver, CANADA, 2019: pp. 9737–9748.

[9] C. Zhao, A. Vij, S. Malhotra, J. Tang, H. Tang, D. Pienta, Z. Xu, W. Zhou, Automatic extraction and stenosis evaluation of coronary arteries in invasive coronary angiograms, Comput. Biol. Med. 136 (2021) 104667. <https://doi.org/10.1016/j.compbiomed.2021.104667>.

[10] H. Zhang, Z. Gao, D. Zhang, W.K. Hau, H. Zhang, Progressive Perception Learning for Main Coronary Segmentation in X-ray Angiography, IEEE Trans. on Med. Imaging. 42 (2022): 864-879. https://doi.org/10.1109/TMI.2022.3219126.

[11] M. Zhai, T. Du, R. Yang, H. Zhang, Coronary Artery Vascular Segmentation on Limited Data via Pseudo-Precise Label, in: Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2019: pp. 816–819. <https://doi.org/10.1109/EMBC.2019.8856682>.

[12] C. Zhao, R. Bober, H. Tang, J. Tang, M. Dong, C. Zhang, Z. He, M. Esposito, Z. Xu, W. Zhou, Semantic Segmentation to Extract Coronary Arteries in Invasive Coronary Angiograms, J. Adv. Appl. Comput. Math. 9 (2022) 76–85. https://doi.org/10.15377/2409-5761.2022.09.6.