

Geometric Learning-Based Transformer Network for Estimation of Segmentation Errors

Sneha Sree¹, Mohammad Al Fahim¹, Keerthi Ram², and Mohanasankar
Sivaprakasam^{1,2}

¹ Indian Institute of Technology Madras, India

² Healthcare Technology Innovation Centre, IIT Madras, India
{snehacumsali,alfahimmohammad}@gmail.com, keerthi@htic.iitm.ac.in,
mohan@ee.iitm.ac.in

Abstract. Many segmentation networks have been proposed for 3D volumetric segmentation of tumors and organs at risk. Hospitals and clinical institutions seek to accelerate and minimize specialists' efforts in image segmentation, but in case of errors generated by these networks, clinicians would have to edit the generated segmentation maps manually.

Problem Statement: Given a 3D volume and its putative segmentation map, we propose an approach to identify and measure erroneous regions in the segmentation map. Our method can estimate error at any point or node in a 3D mesh generated from a possibly erroneous volumetric segmentation map, serving as a Quality Assurance tool.

Method: We propose a graph neural network-based transformer based on the Nodeformer architecture to measure and classify the segmentation errors at any point. We have evaluated our network on a high-resolution μ CT dataset of the human inner-ear bony labyrinth structure by simulating erroneous 3D segmentation maps. Our network incorporates a convolutional encoder to compute node-centric features from the input μ CT data, the Nodeformer to learn the latent graph embeddings, and a Multi-Layer Perceptron (MLP) to compute and classify the node-wise errors.

Results: Our network achieves a mean absolute error of ~ 0.042 over other Graph Neural Networks (GNN) and an accuracy of 79.53% over other GNNs in estimating and classifying the node-wise errors, respectively. We also put forth vertex-normal prediction as a custom pretext task for pre-training the CNN encoder to improve the network's overall performance. Qualitative analysis shows the efficiency of our network in correctly classifying errors and reducing misclassifications.

Keywords: 3D Segmentation error detection, geometric learning

1 Introduction

Medical image segmentation is crucial to isolate and analyze specific structures or regions of interest in a medical image to aid in the diagnosis, treatment planning, and monitoring of diseases or conditions. Deep learning models have

evolved in accuracy, versatility, and deployment-readiness for automatic segmentation of various organs across diverse medical imaging modalities [13], [14], [4]. Still, automated medical image segmentation needs output review, as models are known to be overconfident, although dealing with natural biological variations and diversity in pathological presentation. There is a need for an automated method of predicting and identifying segmentation errors to aid in improving the segmentation maps in erroneous regions.

Related Works: Many recent works have studied the problem of detecting segmentation errors. Kronman et al. [10] proposed a geometrical segmentation error detection and correction method in which they detect segmentation errors by casting rays from the interior of the initial segmentation map to its outer surface. Altman et al. [2] created an automatic contour quality assurance method that utilizes a knowledge base of historical data. Chen et al. [3] proposed supervised geometric attribute distribution models to identify contour errors accurately. The Reverse Classification Accuracy method [12] identifies failed segmentations to predict the CMRI segmentation metrics, achieving a strong correlation with the predicted metrics and visual quality control scores. Alba et al. [1] utilized a random forest classifier with statistical, pattern, and fractal descriptors to detect segmentation contour failures directly without the need for intermediate regression of segmentation accuracy metrics. Roy et al. [15] presented an approach that directly incorporates a quality measure or prediction confidence within the segmentation framework. This measure is derived from the same model, eliminating the need for a separate model to evaluate quality. By leveraging model uncertainty, their approach avoids the requirement of training an independent classifier for evaluation, which could introduce additional prediction errors.

Graph Neural Networks (GNN) are deep learning algorithms that can extract features from complex graph structures through message-passing. They are particularly suited for processing three-dimensional data and extracting geometric features to capture and analyze the data structure [18]. Henderson et al. [9] proposed a quality assurance tool for identifying segmentation errors in 3D organs-at-risk (OAR) segmentations using a geometric learning method by considering the parotid gland. Their study focuses on the parotid gland in head-neck CT scans.

Inspired by this work [9], we propose a novel segmentation error identification network to predict and classify segmentation errors in the inner ear human bony labyrinth using Nodeformer [20], an advanced Transformer based GNN. We also investigate the effect of pre-training tasks on improving the encoding of node feature vectors for GNNs. The key contributions of our work are: **(1)** We propose a novel 3D segmentation error estimation network based on graph learning, capable of handling graphs with millions of nodes generated from 3D segmentation maps. **(2)** We present VertNormPred, a novel pretext task for pre-training the encoder of our network. It involves predicting the node-wise vertex normals to capture the graph’s geometric relationships and surface orientations. **(3)** We quantitatively and qualitatively evaluate our network against other GNN models to estimate and classify node-wise segmentation errors.

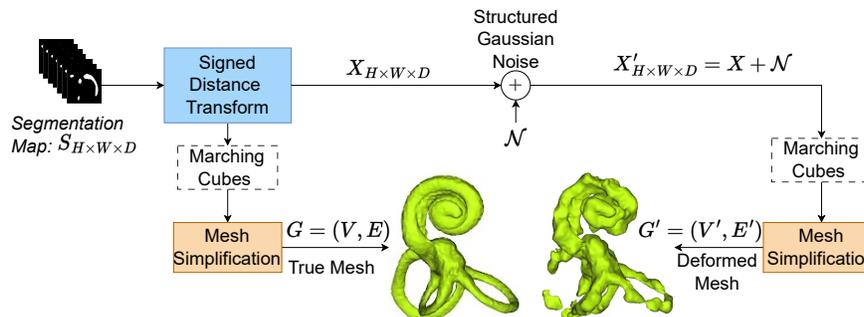


Fig. 1. Simulation of true and perturbed meshes for self-supervised learning of segmentation errors. Each specimen’s SDT was perturbed 100 times to produce 100 different deformed segmentations and meshes. The mesh simplification process utilized Taubin smoothing and quadric error decimation techniques to achieve smoother mesh representations.

2 Methods

Formulation: Let S be the input segmentation map of the μ CT volume I . Let T be the true segmentation map of I . There exists a deformation on S which operates in the voxel grid to transform S to approach T (limited to nearest neighbors interpolation). The alternative (and finer) domain for mutating S is the surface mesh, computed through a discrete Marching Cubes algorithm [11] f on (per-label) extracted contours.

By defining contours as zero-crossings on a signed Euclidean Distance Transform, we have an additional interim domain of the distance transform, which though residing on the voxel grid, offers some unique properties. For instance, take $X' = SDT(S)$ to be the Signed Distance Transform of S , and likewise, for the true segmentation map, define $X = SDT(T)$. A dense deformation mapping S to T is modeled conveniently as an additive distortion of X with a structured (sparse) ‘noise’ field: $X' = X + \mathcal{N}$, and recovering T from S becomes estimating and subtracting the noise in X' . Further, the discrete distance transform domain can be interpolated to match the resolution of the surface mesh.

Thus, estimating a per-voxel additive correction on X' , conditioned on I would lead to determining the location and magnitude of errors in segmentation. This is mapped to learning from ground truth segmentations T through known random perturbations applied in the form $X' = SDT(T) + \mathcal{N}_{sim}$, leading to a self-supervised learning problem, as shown below.

$$\begin{aligned} X &= SDT(T), T = X \leq 0 \\ X' &= X + \mathcal{N}_{sim}, S = X' \leq 0 \end{aligned} \quad (1)$$

Instead of solving this in the SDT domain, we proceed to the mesh domain to setup a per-mesh-vertex estimation of $\mathcal{N}(v)$ conditioned on I , which is equivalent to a corrective field in the interpolated SDT space.

Graph learning: The surface mesh of a segmentation map S , computed through an operation such as the Discrete Marching Cubes, is representable as a graph $G' = (V', E')$, whose nodes are the mesh vertices, and edges the sides of the triangular faces.

$$G' = (V', E') = f(X') \quad (2)$$

A vertex v_i can be localized in the voxel grid of I to assign an interpolated intensity value. Extending further, a local subvolume in I can be defined around v_i . Finally, v_i is connected to nearby vertices forming a local topological arrangement conditioned on image structure. To capture these relationships jointly in the mesh and image domain, we propose to use graph neural networks.

The learning task is the prediction of node-wise segmentation errors by predicting node-wise Signed Distances (SD) and classifying the node-wise SD into different ranges, given the μ CT subvolume centred at each node v , and the entire mesh G' .

The GNN is setup as

$$\hat{\mathcal{N}}(v') := h_{\theta}(G', I) \quad \forall v' \in V' \quad (3)$$

and optimized as

$$\theta^* = \arg \min \left\| \hat{\mathcal{N}} - \mathcal{N}_{sim} \right\|_2^2 \quad (4)$$

Modeling: We propose a graph learning network based on NodeFormer [20], an advanced Transformer based model designed for efficient node classification on large graphs. NodeFormer incorporates an all-pair message-passing method on adaptive latent structures, enabling information exchange between all nodes by effectively capturing the local and global context. To handle larger graphs, Nodeformer employs the kernelized Gumbel-Softmax operator [20], enabling scalability to millions of nodes.

Our intuition behind the model design was, a CNN encoder can capture contextual details from the μ CT data, while the GNN effectively utilizes the local neighborhood of the graph, considering the associated data for each node v . By leveraging the graph’s local neighborhood based on data, the GNN can analyze the relationships and connectivity between graph elements, allowing the model to incorporate both the image contextual information from μ CT data and the geometric structure of the input. This approach enables the model to exploit the information provided by the local neighborhood of each graph element, enhancing its ability to analyze and process the input data effectively.

2.1 Architecture

We choose a CNN consisting of two 3D Conv layers, each followed by ReLU activation functions as a feature extractor to produce node-wise representations

of a $5 \times 5 \times 5$ μ CT subvolume centered around each node. The extracted node features are embedded with the perturbed graph’s edge connectivity information and passed on to the graph transformer network, consisting of three Nodeformer Conv layers. This takes in the graph-embedded node-wise representations and performs all pair message-passing, updating each node’s representation. We consider three Nodeformer Conv layers with eight attention heads, and Batch Normalization and a Leaky ReLU activation function followed each layer. Finally, a Multi-Layer Perceptron (MLP) consisting of three fully connected layers, wherein each layer was followed by a ReLU activation function, Batch Normalization, and a Dropout regularization, processes the updated node-wise representations to produce node-wise SD predictions (using Tanh activation function in the last layer) or classifications (using Softmax activation function in the last layer).

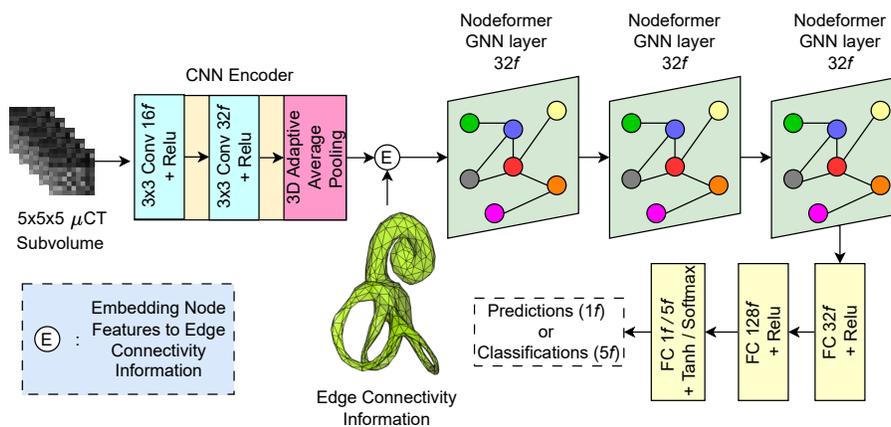


Fig. 2. The proposed graph learning-based transformer network for predicting and classifying node-wise errors. $\#f$ represents the number of output channels/nodes. Given the $5 \times 5 \times 5$ μ CT subvolume centred at each node and the edge connectivity information of the perturbed mesh, the model predicts the errors at each node.

For classification, predicted node-wise SDs are classified into five classes as shown in Fig. 5, ranging from SDs of -0.16 mm to $+0.16$ mm. Nodes falling into the higher end of the range, exceeding $+0.16$ mm, suggest the occurrence of out-segmentation errors in broad regions. Conversely, nodes with SDs below -0.16 mm indicate in-segmentation errors specifically within narrow regions. These observations highlight the correlation between SDs and the likelihood of realistic segmentation errors in different regions of interest. Fig. 2 illustrates the proposed network architecture for node-wise SDs prediction and classification.

2.2 Pre-training Tasks

Towards improving the prediction of node-wise SDs, we incorporated the pre-training transfer learning technique by initializing the model with pre-trained weights obtained from training on different pretext tasks. This approach allows leveraging the knowledge and representations learned during the pretext task to tackle the mainstream tasks [21].

We considered the following three pretext tasks: our custom 1) Vertex Normal Prediction (VertNormPred), 2) μ CT volume Reconstruction (ReconCT), and 3) Masked μ CT volume Reconstruction (MaskReconCT) tasks. In the VertNormPred task, we train the CNN model shown in Fig. 3(a) to predict the node-wise vertex normal X_{vn} given the $5 \times 5 \times 5$ μ CT subvolume centred around a node. While generating the dataset using the marching cubes algorithm, we also obtained the ground truth node-wise vertex normals for each mesh. This task enabled the model to capture geometric relationships and surface orientations. Since neighboring nodes and their orientations influence node-wise SDs [5], understanding surface properties through vertex normal prediction significantly improved the accuracy of the SDs predictions.

In the ReconCT task, we train an encoder-decoder network illustrated in Fig. 3(b) to reconstruct $5 \times 5 \times 5$ μ CT subvolumes. This task allowed the CNN encoder to extract essential features from the node-wise μ CT data.

In the MaskReconCT [8] task, we focus on reconstructing pixel-wise randomly masked $5 \times 5 \times 5$ μ CT subvolumes using an encoder-decoder network shown in Fig. 3(b). We train the model to infer missing regions in the data. By

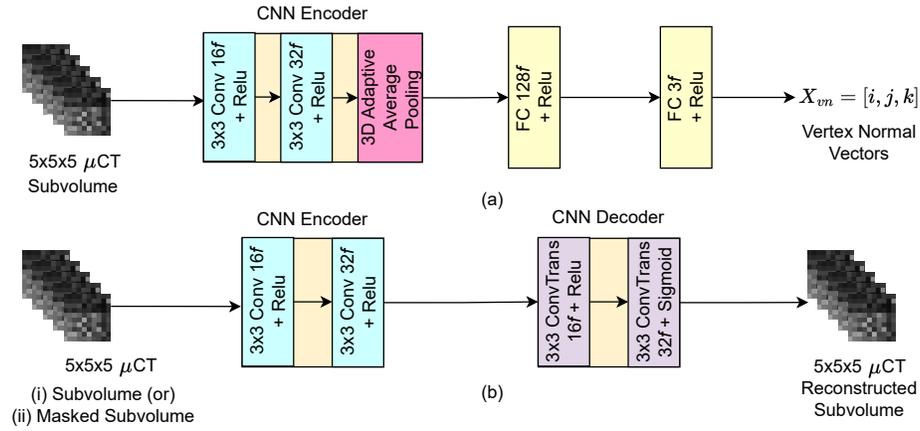


Fig. 3. a) Vertex Normal Prediction (VertNormPred) network predicts the node-wise vertex normals given the $5 \times 5 \times 5$ μ CT subvolume centred at each node. b) (i) CT volume Reconstruction (ReconCT) network and (ii) Masked CT volume Reconstruction (MaskReconCT) network reconstructs the $5 \times 5 \times 5$ μ CT subvolume given μ CT or pixel-wise randomly masked μ CT subvolume centred at each node, respectively.

learning to fill these gaps, the model becomes more adept at estimating SDs, especially when parts of the μ CT are incomplete.

We initialized the CNN encoder of our model with the pre-trained weights obtained from the CNN encoder of the models shown in Fig. 3 from these pretext tasks to facilitate node-feature extraction. The pretext tasks: VertNormPred, ReconCT, and MaskReconCT, improved the model in capturing the μ CT bony labyrinth structure for the mainstream task of prediction/classification of node-wise SD.

3 Dataset Description

We use the publicly available OpenAIRE’s human bony labyrinth dataset [19] to evaluate the method. The dataset consists of clinical Computed Tomography (CT) volumes, co-registered high-resolution micro-CT (μ CT) volumes, segmentation maps, and surface models of 23 human bony labyrinths. We used 22 specimens of μ CT volumes and their corresponding segmentation maps.

3.1 Generation of Training Data

We generate the perturbed segmentation maps by perturbing the SDT by addition of noise of the true segmentation map 100 times, ensuring the Hausdorff distances of the perturbed segmentation maps are in the range of (7 – 65). Fig. 4 illustrates the simulation of a perturbed segmentation map obtained from a perturbed SDT.

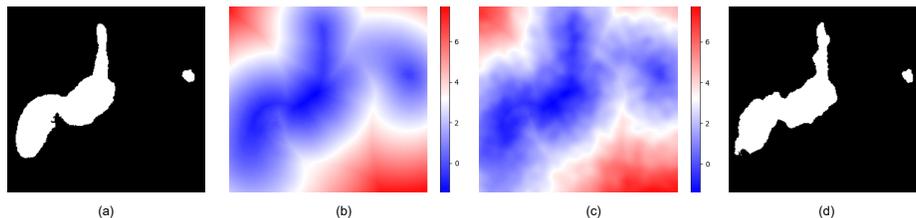


Fig. 4. One of the slices of (a) true segmentation map, (b) distance transform, (c) perturbed distance transform after addition of noise to distance transform and (d) perturbed segmentation map obtained from the perturbed distance transform (c)

We use the marching cubes algorithm to obtain the triangular mesh manifolds of the perturbed segmentation maps. The complex geometry of the human bony labyrinth led to generating a mesh with numerous triangles, resulting in a graph with nodes in the order of 10^5 . We use Taubin smoothing [16] and quadratic error decimation techniques to smoothen the mesh. We consider the mesh vertices as nodes (V) of the graph and the sides of the triangular faces of the mesh as edges (E). The simulation of true and deformed mesh is shown in Fig. 1.

To calculate the node-wise SD, we perform bi-linear interpolation between the nodes of the perturbed mesh and the voxels of the ground truth SDT. Note that the generated node-wise errors correspond to the node-wise SDs of the true segmentation. For classification, we split these node-wise SDs into five classes ranging from -0.16mm to $+0.16\text{mm}$.

4 Experiments and Results

Towards fine-grained prediction of node-wise SDs, we trained and evaluated our model for regression of node-wise SDs. To also identify the errors in different ranges, we trained and evaluated our model for classification to classify the predicted node-wise SDs into different classes.

For all the experiments, we considered 1400 perturbed structures for training, 200 for validation, and 600 for testing.

We have quantitatively and qualitatively evaluated our model against Spline Conv [7] and GAT [17] based GNN models for regression and classification of node-wise SDs. We have also evaluated the models using pre-trained weights

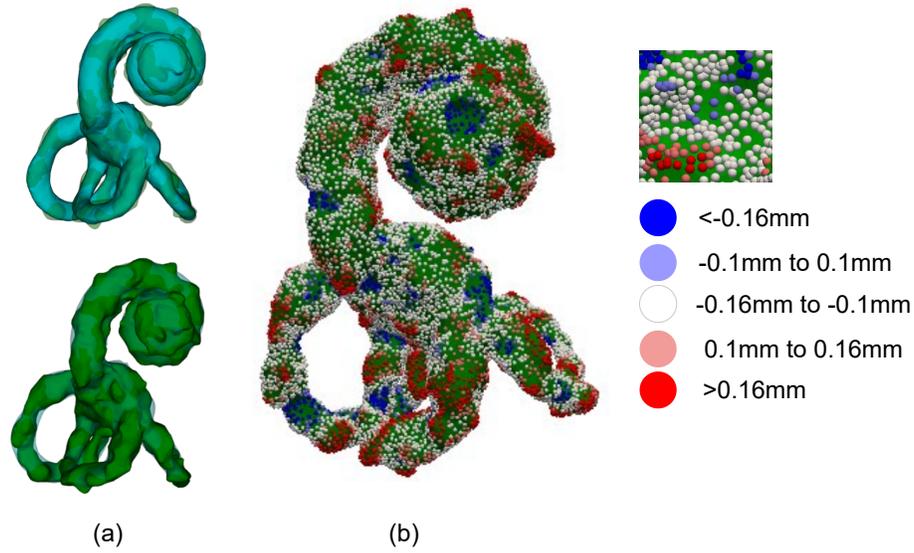


Fig. 5. Visualization of the true, perturbed meshes, and the node-wise SD classes. (a) At the top, the true mesh (in blue) is overlaid with the perturbed mesh (in green). At the bottom, the perturbed mesh (in green) is overlaid with the true mesh (in blue). The overlapping region between the true and perturbed meshes reveals where internal and external segmentation errors occur. (b) The node-wise SDs in the perturbed mesh are distributed into five classes indicated by colours varying from red to blue, and the class ranges are shown above.

Table 1. Comparison of Nodeformer with different pre-trained weights against other models for regression of node-wise SDs.

GNN	Pretraining	MAE ↓	MSE ↓
Spline	ReconCT	0.06994	0.00986
GAT	-	0.06946	0.00913
GAT	VertNormPred	0.0694	0.00968
Spline	MaskReconCT	0.06783	0.00802
GAT	ReconCT	0.06755	0.00884
GAT	MaskReconCT	0.06705	0.00903
Spline	-	0.06032	0.00762
Spline	VertNormPred	0.05728	0.00757
Nodeformer	-	0.04536	0.00475
Nodeformer	MaskReconCT	0.04397	0.00451
Nodeformer	ReconCT	0.04254	0.00444
Nodeformer	VertNormPred	0.04182	0.00429

from the three pretext tasks. We also performed ablation studies to understand the contribution of each block in our proposed model.

4.1 Implementation details

We train the network in Fig. 3(a) for the VertNormPred task, where we minimize the *Cosine Similarity loss* between the predicted and ground truth node-wise vertex normals. We train the network in Fig. 3(b) for ReconCT and MaskReconCT tasks, where we minimize the *L1 loss* between the generated and original $5 \times 5 \times 5$ μ CT node-wise subvolumes.

For regression of node-wise SDs, we train the models to minimize the *Smooth L1 loss* between the predicted node-wise SDs and the node-wise SDs obtained using interpolation (GT SDs). We used the Mean Absolute Error (MAE) and Mean Square Error (MSE) metrics to quantify the performance of the models trained for regression. For the classification of node-wise SDs, we train the models to minimize the *Cross Entropy loss* between the predicted and GT SD classes. We used the F1 score, Precision, Recall, and Accuracy metrics to quantify the performance of the models trained for classification. We trained all the networks for 100 epochs, using a learning rate of $1e^{-3}$ and a cosine annealing scheduler with a weight decay of $1e^{-3}$. Both the regression and classification models utilized the AdamW optimizer, while the pre-training networks employed the Adadelta optimizer. Models are implemented using PyTorch and PyG [6], and the training process was carried out in a workstation using an i5-1035G4 CPU and NVIDIA 24GB RTX 3090 GPU.

4.2 Results and Discussion

In Table 1, it can be observed that our model built upon Nodeformer can predict node-wise SDs efficiently, and additionally, using pre-trained weights improved

the prediction. Among the evaluated models, our model with a CNN encoder initialized with VertNormPred pre-trained weights yielded the lowest MAE score of 0.04182. This signifies a substantial improvement of $\sim 30.6\%$ compared to Spline Conv GNN without any pre-trained weights.

Table 2. Comparison of Nodeformer with different pre-trained weights against other models for classification of node-wise SD classes.

GNN	Pretraining	f1 Score \uparrow	Precision \uparrow	Recall \uparrow	Accuracy(%) \uparrow
Spline	MaskReconCT	0.4872	0.5445	0.5124	66.3
GAT	VertNormPred	0.5186	0.605	0.5398	69.03
GAT	MaskReconCT	0.5024	0.5649	0.5425	71.22
Spline	VertNormPred	0.5367	0.612	0.5487	71.76
GAT	ReconCT	0.5746	0.6289	0.5927	72.17
Spline	ReconCT	0.5871	0.6136	0.614	72.28
GAT	-	0.5623	0.6487	0.567	72.4
Spline	-	0.5582	0.6181	0.5779	71.53
Nodeformer	MaskReconCT	0.5986	0.6693	0.589	74.55
Nodeformer	ReconCT	0.6695	0.72	0.7131	76.57
Nodeformer	-	0.6899	0.7343	0.6693	78.82
Nodeformer	VertNormPred	0.6943	0.7384	0.6835	79.53

In Table 2, our model, initialized with pre-trained encoder weights of the VertNormPred task, gave an overall accuracy of 79.53%. This signifies a substantial improvement of $\sim 8\%$ in accuracy compared to the Spline Conv GNN without any pre-training task, indicating a significant improvement in the model’s ability to identify different ranges of segmentation errors.

Tables 1 and 2 show that our model has benefited from using the pre-trained weights of the VertNormPred task, indicating that the prediction of the node-wise vertex normals during pre-training has helped the encoder of our model in capturing the intricate surface orientations and geometric inter-node relationships in the bony labyrinth structure. This has helped further improve the prediction of node-wise SDs.

From Fig. 6, it is evident that our model, using Nodeformer, outperforms the other models in the classification of node-wise SDs. Our model qualitatively exhibits improved classification of node-wise SD classes, with significantly fewer black-colored nodes representing incorrect predictions than those obtained using Spline Conv and GAT models. This highlights our model’s superior performance and effectiveness in accurately classifying the node classes in the given graph.

In our experiments, we observed that incorporating pre-trained weights from the pretext tasks positively impacted the performance of the models in the regression of node-wise SDs. However, using pre-trained weights for the classification task did not result in much significant improvement.

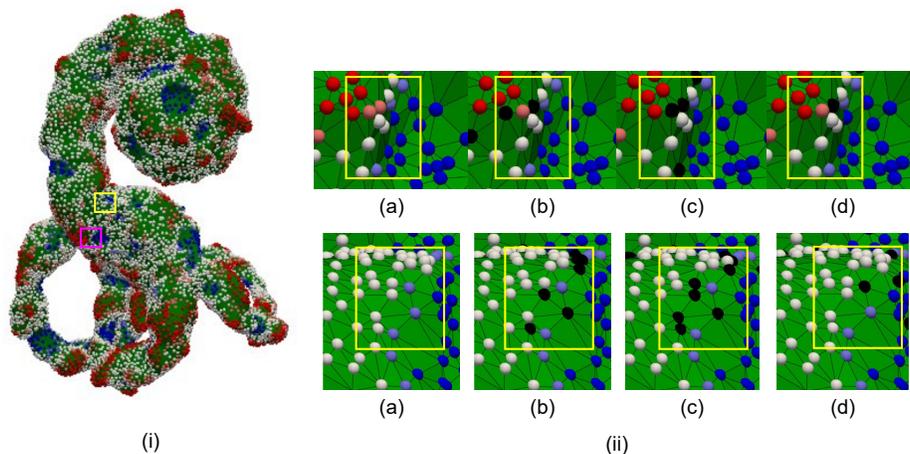


Fig. 6. Visual illustration of the classification of node-wise SDs by all the models. (i) The actual node-wise SD classes in the perturbed mesh. (ii) Two distinct regions within the graph to showcase the predicted node classes compared to the ground truth node classes. The first and second rows show the zoomed-in regions in the pink and yellow boxes in the perturbed mesh (i). (a) GT perturbed node-wise SD classes, (b)-(d) predicted node-wise SD classes by Spline Conv, GAT, and our model, respectively. The black-coloured nodes denote incorrect predictions. The yellow boxes in (ii) (a)-(d) show how well our model can classify the node-wise SD classes with respect to the GT node-wise SD classes with the least number of black nodes.

4.3 Ablation Study

To evaluate the extent to which Nodeformer effectively learns meaningful information from the geometric structure of the segmentation, we performed an ablation study for the classification task that involved removing the GNN component entirely and directly passing on the node-wise representations from the encoder to the MLP decoder. Also, to evaluate the importance of node feature extraction using the CNN encoder and pre-trained weights, we experimented by passing the μ CT subvolumes through a linear layer as node feature representations to the Nodeformer instead of passing them through the CNN encoder.

Fig. 8 demonstrates the significance of incorporating geometrical structure learning using Nodeformer and the CNN encoder to extract node features in identifying segmentation errors by comparing their performance in classification. Upon removing Nodeformer from our model (CNN-MLP), the classification performance for error identification is notably poor. This emphasizes the importance of Nodeformer in capturing the geometrical information required for error analysis.

Furthermore, using a linear layer to extract node features from μ CT subvolumes instead of the CNN encoder also resulted in poor performance, as shown in Fig. 8. This highlights the importance of Conv layers in effectively capturing the node-centred μ CT information necessary for accurate error classification.



Fig. 7. Confusion Matrices of (a) CNN-MLP: Node features from the CNN encoder are directly given to the MLP for classification, (b) GNN-MLP: Node feature vectors are obtained from a linear layer instead of the CNN encoder are passed on to the Nodeformer and MLP for classification, (c) our complete model, and (d) our model with the CNN encoder initialized with the VertNormPred pre-trained weights. Labels A: ($<-0.16\text{mm}$), B: (-0.16mm to -0.1mm), C: (-0.1mm to 0.1mm), D: (0.1mm to 0.16mm), E: ($>0.16\text{mm}$)

Regarding using pre-trained weights, our model with the CNN encoder initialized with pre-trained weights from the VertNormPred task gave the best classification performance regarding accuracy, precision, recall, and F1 score, as shown in Figs. 7 and 8.

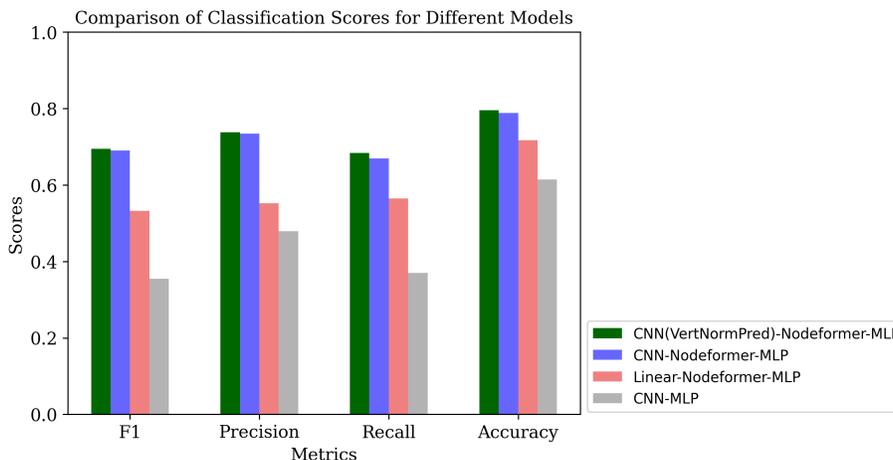


Fig. 8. Comparison of Classification Scores between the different blocks of our model, described in section 4.3. The plot provides a visual representation of the distribution and relative performance of the models based on their classification scores.

By comparing Fig. 7(a) and Figs. 7(b)-(d), it can be observed that removing the GNN component (Nodeformer) led to a notable decrease in the model’s performance in classifying errors in different ranges. Specifically, it fails to identify internal errors (recall score of 7.6%). Fig. 7(b) shows that the Linear-Nodeformer-MLP (GNN-MLP) model can identify internal and external errors but fail to identify the intermediate ones. From Fig. 7(c) and (d), it is clear that the models using Nodeformer for geometrical structure learning with a CNN encoder for node feature extraction were capable of identifying errors in all ranges and using pre-trained weights reduced misclassification in some classes and significantly improved the recall score.

5 Conclusion

Our work introduced a Nodeformer-based graph learning network as a Quality Assurance (QA) tool to evaluate errors in the automatic segmentation of medical images. To our knowledge, this is the first work that addresses segmentation errors in the 3D data of the human inner-ear bony labyrinth structure. The complexity of the inner-ear human bony labyrinth structure gave rise to graphs with nodes in the order of 10^5 . Our network, built upon Nodeformer, can scale up to millions of nodes and easily handle human inner-ear bony labyrinth graphs. To boost the performance of our network, we also proposed a custom Vertex Normal Prediction pretext task for pre-training the CNN encoder of our network. We have evaluated our network against other GNN models with pre-trained weights from different pretext tasks for regression and classification of node-wise segmentation errors. We have qualitatively shown how well our model

can correctly classify segmentation errors and reduce misclassifications. We have also conducted an ablation study to show the strengths of individual modules of our network, along with loading the pre-trained weights from the Vertex Normal Prediction pretext task, for classification. This study motivates further research into developing and advancing QA techniques and tools for measuring, classifying, and correcting segmentation errors.

References

1. Alba, X., Lekadir, K., Pereanez, M., Medrano-Gracia, P., Young, A.A., Frangi, A.F.: Automatic initialization and quality control of large-scale cardiac mri segmentations. *Medical image analysis* **43**, 129–141 (2018)
2. Altman, M., Kavanaugh, J., Wooten, H., Green, O., DeWees, T., Gay, H., Thorstad, W., Li, H., Mutic, S.: A framework for automated contour quality assurance in radiation therapy including adaptive techniques. *Physics in Medicine & Biology* **60**(13), 5199 (2015)
3. Chen, H.C., Tan, J., Dolly, S., Kavanaugh, J., Anastasio, M.A., Low, D.A., Harold Li, H., Altman, M., Gay, H., Thorstad, W.L., et al.: Automated contouring error detection based on supervised geometric attribute distribution models for radiation therapy: a general strategy. *Medical physics* **42**(2), 1048–1059 (2015)
4. Chen, Y., Ruan, D., Xiao, J., Wang, L., Sun, B., Saouaf, R., Yang, W., Li, D., Fan, Z.: Fully automated multiorgan segmentation in abdominal magnetic resonance imaging with deep neural networks. *Medical physics* **47**(10), 4971–4982 (2020)
5. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5939–5948 (2019)
6. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: *ICLR Workshop on Representation Learning on Graphs and Manifolds* (2019)
7. Fey, M., Lenssen, J.E., Weichert, F., Müller, H.: Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 869–877 (2018)
8. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 16000–16009 (2022)
9. Henderson, E.G., Green, A.F., van Herk, M., Vasquez Osorio, E.M.: Automatic identification of segmentation errors for radiotherapy using geometric learning. In: *Medical Image Computing and Computer Assisted Intervention—MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part V*. pp. 319–329. Springer (2022)
10. Kronman, A., Joskowicz, L.: A geometric method for the detection and correction of segmentation leaks of anatomical structures in volumetric medical images. *International journal of computer assisted radiology and surgery* **11**, 369–380 (2016)
11. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353 (1998)
12. Robinson, R., Valindria, V.V., Bai, W., Oktay, O., Kainz, B., Suzuki, H., Sanghvi, M.M., Aung, N., Paiva, J.M., Zemrak, F., et al.: Automated quality control in image segmentation: application to the uk biobank cardiovascular magnetic resonance imaging study. *Journal of Cardiovascular Magnetic Resonance* **21**(1), 1–14 (2019)

13. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)
14. Roth, H.R., Shen, C., Oda, H., Oda, M., Hayashi, Y., Misawa, K., Mori, K.: Deep learning and its application to medical image segmentation. *Medical Imaging Technology* **36**(2), 63–71 (2018)
15. Roy, A.G., Conjeti, S., Navab, N., Wachinger, C., Initiative, A.D.N., et al.: Bayesian quicknat: Model uncertainty in deep whole-brain segmentation for structure-wise quality control. *NeuroImage* **195**, 11–22 (2019)
16. Taubin, G.: Curve and surface smoothing without shrinkage. In: Proceedings of IEEE international conference on computer vision. pp. 852–857. IEEE (1995)
17. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al.: Graph attention networks. *stat* **1050**(20), 10–48550 (2017)
18. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)* **38**(5), 1–12 (2019)
19. Wimmer, W., Anschuetz, L., Weder, S., Wagner, F., Delingette, H., Caversaccio, M.: Human bony labyrinth dataset: Co-registered ct and micro-ct images, surface models and anatomical landmarks. *Data in brief* **27**, 104782 (2019)
20. Wu, Q., Zhao, W., Li, Z., Wipf, D.P., Yan, J.: Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems* **35**, 27387–27401 (2022)
21. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? *Advances in neural information processing systems* **27** (2014)