Generative modeling for shapes using graph convolutional networks (generation)

Tan Jian Sean

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

6th September 2023

Abstract

Machine learning or artificial intelligence models have been widely received in the medical field since their introduction. These models are often trained using a population of real patient data and require a large set of training data to achieve the desired performance. However, privacy concerns can hinder healthcare providers from sharing their repository of patient data for deep learning or machine learning applications. With the help of generative models, we can produce synthetic human body organ shapes that resemble the actual data distribution. This project is interested in molding surface meshes as graphs and developing a framework of neural networks that can be trained through an unsupervised approach using this form of graph data to tackle the issue mentioned. This paper attempts to apply a modified version of the ASMG framework presented by Kalaie *et al* [21] to the AMOS dataset [30] to create synthetic abdominal organ shapes. While the 3D meshes produced did not achieve good performance, future enhancements that could improve the proposed method are discussed in detail.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in electronic form.

Name: Tan Jian Sean

Signature: [illegible]

Acknowledgements

I would like to thank my family who supported me throughout my study both financially and mentally. I would also like to thank Dr. Ali Gooya for his guidance on very important directions and concepts for this project. Finally, I would like to thank all my classmates, lecturers, and lab assistants, especially those whom I worked with and received help from. Finally, even though this project was not as complete as I would have liked, I have given my best effort and for that, I would like to thank myself.

Contents

Chapter 1 Introduction

1.1 Motivation

Machine learning or artificial intelligence models have been widely received in the medical field since their introduction. One of the key factors is that relationships between data of interest are often straightforward, which allows understanding and explanation of relationships between such data [1]. Such understanding of data and algorithms is especially crucial in a field like medicine where ethics and safety are of the utmost importance.

AI models are often trained using a population of real patient data to assist with computer-aided diagnosis, risk stratification, and prediction of disease trajectories. These models require a large set of training data to achieve the desired performance. However, despite the implementation of different regulations and standards on medical records [2], privacy concerns can still hinder healthcare providers from sharing their repository of patient data for deep learning or machine learning applications. With the help of generative models, we can produce synthetic human body organ shapes that resemble the actual data distribution. Numerous AI models can be the beneficiary of such generated shapes when used in the training process.

1.2 Aims and Objectives

This project is interested in molding surface meshes as graphs and developing a framework of neural networks that can be trained through an unsupervised approach using this form of graph data to generate synthetic shapes which resemble the actual training shapes.

The high-level objectives for the project are:

1) Study the latest state-of-the-art implementation of Graph Neural Networks and Shape Generation
2) Preprocess the chosen dataset to suit the needs of the Graph Neural Network
3) Implement a framework that can take an unstructured graph as input and produce the final output of synthetic shapes that resemble the distribution of the input data.

Chapter 2 Background

2.1 Graph Neural Networks

The accelerated development in the deep learning community sees the inventions and applications of various deep learning paradigms. These inventions are designed to deal with different types of data. For example, convolutional neural networks (CNNs) for image data, recurrent neural networks (RNNs) for temporal data and autoencoders (AEs) for feature extraction.

However, the paradigms mentioned above, while performing well against regular data like images, text, or videos, struggle with data whose latent representation cannot be extracted from Euclidean data [3]. Graph data is a good example with a lot of practical use cases across several industries, including but not limited to molecules study, network study and closely related to this project, 3D data. The irregular nature of graph data brings difficult challenges to the aforementioned machine learning algorithms as a graph can have a changing number of unordered nodes, and the nodes can have a different number of neighbours.

Modern studies have taken on the challenge by coming up with Graph Neural Networks (GNNs). The intuition is to generalize and redefine the important operations in the deep learning algorithms (e.g., convolutions and pooling) to enable the processing of said graph data. According to the taxonomy of GNNs shown in [3], GNNs can be categorized into 4 types: 1) Recurrent GNNs (RecGNNs), 2) Convolutional GNNs (ConvGNNs), 3) Graph Autoencoders (GAEs), and 4) Spatial-Temporal GNNs (STGNNs).

RecGNN works are considered the earlier works of GNNs. The first few research centred around directed acyclic graphs to compensate for the lack of available computational power [3] while following RecGNN implementations allows the processing of more general types of graphs [4]. RecGNNs generally adopt a contraction mapping recurrent function to guarantee convergence when trying to extract high-level node representation.

2.2 Convolutional Graph Neural Networks

Spectral-based ConvGNNs have a strong mathematical foundation from the field of Graph Signal Processing (GSP) [5] where they require the eigen-decomposition of the graph Laplacian. Defferrard *et al* [6] pioneered the field with the implementation of ChebNet. They generalized convolutions to graph data using a spectral convolution, along with the introduction of a pooling method based on graph coarsening. Multiple further studies on the same approach are built on

ChebNet to improve the calculation efficiency, including FastGCNs [7], SGCs [8] and CayleyNets [9].

On the other hand, spatial-based ConvGNNs' definition of graph convolutions is based on the node's spatial relations. The design of spatial-based convolutions makes sure that the central node's new representation is derived from its representation along with each of its connected nodes' representation via message passing, similar to RecGNNs. Overall, spatial ConvGNNs have been gaining popularity due to their applicability towards other neural networks. GraphSage by Hamilton *et al* [10] utilizes fixed neighbourhood sampling and aggregation techniques to achieve impressive performance in an efficient manner. GraphSage is able to make predictions of the embeddings from unseen nodes, making it a well-regarded choice when working with large graphs. Mixture Model Networks (MoNet) by Monti *et al* depicts the local neighbourhoods of nodes using mixture models. This approach generalizes several studies while integrating new techniques like pseudo-coordinates and parametric kernel, resulting in a framework that has useful characteristics like parameter sharing, Gaussian kernel with learnable weight and possibility for transfer learning, making it a general framework that can be applied to a variety of non-Euclidean domain (e.g., graphs and manifolds).

2.2.1 Graph Attention Network

Graph attention network (GAT) [12] tries to deal with situations where different neighbour nodes should have different weights. It managed to do so by utilizing masked self-attention layers, allowing the framework to assign different weights to different neighbours without knowing the entire graph structure upfront. This results in a framework that is able to learn node embeddings that are sensitive to the graph structure which is especially effective against node classification tasks.

2.2.2 Feature-Steered Graph Convolutions for 3D Shape Analysis

Feature-steered graph convolutions for 3D Shape Analysis (FeaStNet) is a GNN based on a novel graph convolution operator presented in [17]. The operator allows the network to dynamically adjust local filter weights according to the embeddings learned in the previous layer, which opens up new possibilities compared to earlier studies where the graph convolution has fixed weights. The model has been tested and has shown robust capability when dealing with regular or irregular 3D shape analysis tasks.

2.3 Graph Autoencoders (GAEs)

Graph Autoencoders are essentially Autoencoders that are designed to deal with graph data. The overall structure is similar to a traditional autoencoder, which consists of an encoder and a decoder. The decoder maps graph nodes into a latent space while the decoder reconstructs a graph from the learned representations in the latent space.

GAEs produce node embedding, which is a low-dimensional vector representation of a node. The learned representation should keep the topological information of the original node. The encoder performs compression on the input node to collapse it into a lower dimension while the decoder attempts to reconstruct a graph from the latent representations. The loss function will typically measure the difference between the reconstructed node and the original node, forcing the encoder to construct a useful latent space. Variational GAE (VGAE) [13] is a mutation of GAE that aims to learn the distribution of data by maximizing the likelihood of the data under a probabilistic model, similar to the original Variational Autoencoders (VAEs) [14]. Kalaie *et al* [21] implemented a VGAE network in the Attention-based Shape Matching and Generation (ASMG) framework to perform unsupervised learning on the 3D data in order to capture the vertex-to-vertex correspondences needed for the implementation of an attention mechanism.

Litany *et al* [15] presented a graph-convolutional method designed for shape-completion tasks that work on different types of missing data. During training, the graph convolutional autoencoder learns the latent space of 3D shapes represented as graphs. The model can then perform reconstruction of partial 3D shapes and has shown impressive performance on deformable 3D shapes. Convolutional Mesh Autoencoder (CoMA) [16] processes 3D faces that are represented as meshes and have shown the capability to generate extreme non-linear facial expressions, unseen in previous applications of human face generation. The compact framework requires fewer model parameters than alternative options and features a mesh sampling operation that maintains the topological information of the 3D input data.

2.4 β-VAE

Preceding the deep-learning era, PCA-based approaches have been deployed in these shape generations studies, as seen in [27-28]. Recent years have seen deep learning approaches gaining more attention [29]. Inspired by the original VAE [14], ß-VAE [18] aims to learn disentangled representations from the input data in pursuit of finding the representations that are well suited for a specific task or domain. A disentangled representation has single latent units that are sensitive to changes in single generative factors but less so in other factors [19]. Beetz *et al*

[20] proposed a conditional Point Cloud β-VAE, which is a framework that takes in conditional input and performs generation according to the input. Kalaie *et al* successfully generated virtual left ventricles as graphs using β-VAE in the Attention-based Shape Matching Generation framework [21].

Chapter 3 Methodology

3.1 Dataset Description

The dataset used in this project is the AMOS dataset from the MICCAI Challenge 2022 [30], which is a large-scale abdominal multi-organ benchmark for versatile medical image segmentation. The download link for the dataset can be found in *Appendix A.1*. According to the official documentation [22], the dataset consists of 600 CT/MRI (500 CT, 100 MRI) abdominal scans collected from 600 patients. The scans cover a total of 15 organ categories, listed in *Appendix A.2*. The data obtained is in NIFTI archive format (.nii.gz).

This project will focus on CT scans and the organ of the gallbladder. To identify the selected organ, we will make use of the segmented images (training and validation data only) in the label folders. There are 200 training images and 100 validation images (all labelled).

3.2 Data Preprocessing

3.2.1 Extraction & Filtering NIFTI Images

The first step of preprocessing is to extract the relevant slices from the original NIFTI image data containing the target organ. In this step, we will also perform filtering on each of the extracted slices to remove other organs from the relevant slices. All images are resized to the same shape (520, 520, 10). Any extracted slices that have a length not between 4 to 18 are removed to ensure that there are no outliers or null data. The final output from this step is a batch of new uniform trimmed NIFTI images containing only the target organ, derived from the original data.

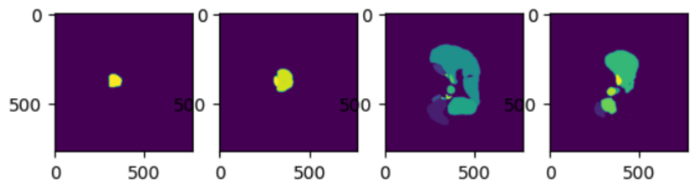


Figure 1: Original NIFTI image slices.

3.2.2 Conversion to 3D mesh

The batches of trimmed NIFTI images are then passed through a function that converts NIFTI files to vtk mesh (.vtk) using the vtkMarchingCubes algorithm. Other methods like vtkFillHolesFilter and vtkSmoothPolyDataFilter are used to further refine the mesh produced. The last step is to convert the vtk mesh to trimesh (.stl) using the vedo library. The meshes are converted to trimesh format in the end as the Pytorch Geometric library supports native conversion of trimesh to its Data object class.

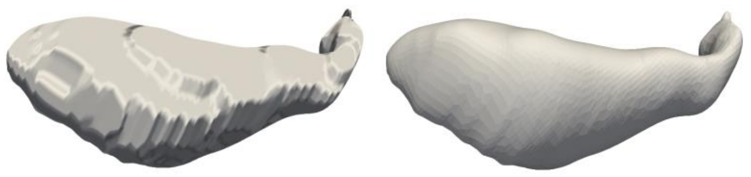


Figure 2: `gallbladder` **meshes, .*vtk*** (left) and .*stl* (right).

3.3 Attention-based Shape Matching and Generation

The framework chosen borrows from the ASMG framework proposed by Kalaie *et al* [21]. This modified ASMG framework consists of (1) an Attention-based Shape Matching (ASM) mechanism which learns a set of vertex-to-vertex correspondences using a VGAE Φ combined with an attention mechanism for domain transformation and (2) a β-VAE framework for the generation of synthetic 3D shapes. The 3D meshes used as input are represented in the form of graph data, allowing the VGAE to perform unsupervised learning on the input data. Figure 3 below shows the entirety of the modified ASMG framework used.

We represent the training set of varying-sized graphs $G = \{g_k\}_{k=1}^{K}$ with the k-th observed graph $g_k = (V_k, E_k)$ containing $|V_k|$ nodes, with E_k representing the set of edges connecting the nodes. The graph g_k data has a node feature matrix $\mathbf{X}_k \in R^{|V_k| \times 3}$ and an adjacency matrix $\mathbf{A}_\mathbf{k} \in R^{2\times|E_k|}$ also known as edge index.

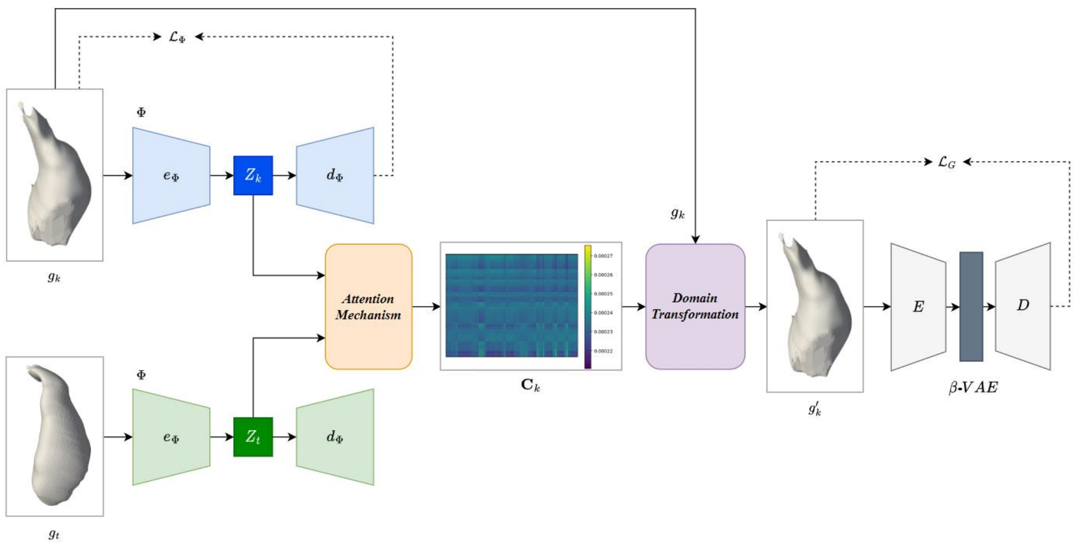


Figure 3: Modified ASMG Framework

3.3.1 Variational Graph Autoencoder

The Variational Graph Autoencoder (VGAE) Φ is built to capture the local and global structural information of an input graph g. The model is first trained in an unsupervised manner where the encoder network e_Φ takes in a set of training input graph g (consisting of its adjacency matrix $\mathbf{A}$ and its node features matrix $\mathbf{X}$) and produces latent space embeddings $\mathbf{Z}$. The decoder network d_Φ then attempts to reconstruct the feature matrix $\mathbf{X}$ (from embedding $\mathbf{Z}$) as $\mathbf{X}^{'}$, which is used for loss feedback for the training process, allowing the VGAE to learn the best nodal representations for input graph g.

The loss function $\mathcal{L}_\Phi$ is a combination of reconstruction loss $\mathcal{L}_{rec}$ and the Kullback-Leibler divergence (KL divergence) D_{KL}:

$$\mathcal{L}_\Phi = \sum_{k=1}^{K} \frac{1}{2} \sum_{n=1}^{|V_k|} \left|\left|x_{kn}^{'} - x_{kn}\right|\right|^2 - D_{KL} \tag{1}$$

where $\mathcal{L}_{rec}$ is representation as Mean Squared Error (MSE), and the KL divergence computes the divergence between the Gaussian prior $\mathcal{N}(0,1)$ and posterior distribution of the latent space.

3.3.2 Attention Mechanism

To perform pair-wise correspondences of all the vertices between each of the k-th observed graph g_k and a chosen template graph g_t, we extracted the nodal embeddings computed by VGAE Φ as $\mathbf{Z_k} = \Phi(\mathbf{X_k}, \mathbf{A_k})$ and $\mathbf{Z}_\mathrm{t} = \Phi(\mathbf{X}_\mathrm{t}, \mathbf{A}_\mathrm{t})$ respectively. By using the following formula, we can obtain the soft correspondence matrix in the form of attention maps:

$$\mathbf{C}_k = Softmax\left(\lambda Z_t Z_k^T\right) \tag{2}$$

where with d_z indicating the dimension of the latent vector z (which is also the hidden dimension of the VGAE Φ, 128 in this case), $Z_k \in R^{|V_k| \times d_z}$, $Z_t \in R^{|V_t| \times d_z}$ and $\mathbf{C}_k \in [0,1]^{|V_t| \times |V_k|}$. The hyperparameter λ is set to $1e-3$.

The combination of VGAE Φ and Attention Mechanism allows the learning of nodal embeddings on populations of inconsistent graphs data. This is done via the spatial graph convolution operator deployed which performs convolution locally on each node. The FeaSt convolution layer [17] dynamically shifts its filter weights based on the features learned in the previous layers of the network instead of using a set of static predefined weights. Following the extraction of the latent representations, the attention mechanism's function is deployed to normalize the inconsistent shapes.

3.4 Domain Transformation and Shape Generation

Domain transformation is performed before the stage of shape generation. This step will ensure that there is a set of normalized shapes to serve as the input for the β-VAE, allowing efficient learning and better generation quality. Domain transformation is achieved by using the attention maps $\mathbf{C}_k$, which can map the original node features matrix of observed graphs $\mathbf{X_k} \in R^{|V_k| \times 3}$ from its original node function space $\mathcal{F}(R^{|V_k|})$ to the template graph's node function space $\mathcal{F}(R^{|V_t|})$. When passing through the attention maps, graph areas with low attention are deemed irrelevant and are multiplied with lower weights whereas graph areas with high attention are multiplied with higher weights. This domain transformation can be represented with the formula below:

$$\mathbf{X}_\mathbf{k}^{'} = \mathbf{C}_\mathbf{k}\mathbf{X}_\mathbf{k} \tag{3}$$

where $\mathbf{X}_\mathbf{k}^{'} \in R^{|V_t| \times 3}$ denotes the node feature matrix of the structurally normalized graphs $g_k^{'}$.

The final step of the framework revolves around a β-VAE model, which takes in the node feature matrix $\mathbf{X}_\mathbf{k}^{'}$ of the transformed and normalized shapes $\{g_k^{'}\}_{k=1}^{K}$, perform learning on the probability density function, and generate a set of synthetic shapes.

The generation loss function $\mathcal{L}_G$ is akin to the loss function of the VGAE $\mathcal{L}_\Phi$ which is a combination of reconstruction loss and KL Divergence D_{KL}. The difference here is that there is a hyperparameter β for D_{KL}. This hyperparameter allows the manipulation of the balance between construction error and latent space quality by emphasizing the learning of disentangled representations. The generation loss can be defined as:

$$\mathcal{L}_\mathrm{G} = \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{|V_t|}\left|\left|D(E(\mathbf{x}_\mathbf{kj}^{'})) - \mathbf{x}_\mathbf{kj}^{'}\right|\right|^2 - \beta D_{KL}, \tag{4}$$

Where the β-VAE is defined as an encoder-decoder pair, denoted as a pair of $\{E, D\}$ networks respectively. The KL Divergence D_{KL} computes the divergence between the Gaussian prior $\mathcal{N}(0,1)$ and posterior distribution of the latent space $E(\mathbf{x}^{'})$.

Chapter 4 Experiments and Results

4.1 Software and Hardware

The project was completed on Python 3.9. A major part of the work used the Pytorch and Pytorch Geometric (PyG) library. The VGAE is built using the FeaStConv graph convolution layer from PyG library.

The system used to implement this project is a Laptop with AMD Ryzen 7 5800H, 32GB of RAM, and an Nvidia GeForce RTX 3060 Laptop GPU.

4.2 Experimental Setup

The dataset chosen is the CT scans from the AMOS dataset mentioned in section 3.1, and after the preprocessing there are 163 training mesh as well as 66 validation mesh.

The VGAE model Φ has an encoder e_Φ with hidden layers of dimensions 64, 64, 128, 128. Each layer is a Pytorch implementation of the FeaStNet [17] convolution operator paired with batch normalization layers. The decoder d_Φ is the mirrored version of the encoder. As mentioned in section 2.2.2, the filter weight of the convolution layers will change dynamically according to the features learned by the previous layer in the network. The model uses an Adam [23] optimizer, with the learning rate set to $1e-3$ and was trained for 200 epochs. The structure of the model Φ can be found in *Appendix B Table 1*.

As for the β-VAE, it has a hidden layer dimension of 512, 256, 768, 128, 64 for the encoder E. The decoder D is again the mirrored version of the encoder. The β-VAE uses a combination of fully connected linear layers and batch normalization, with Leaky ReLU as activation layers. The hyperparameter β is set to $2e-6$. The optimizer is an Adam optimizer with the learning rate set to $1e-3$, and the model was trained for 500 epochs. The structure of the β-VAE can be found in *Appendix B Table 2.*

4.3 Results and Discussion

As demonstrated in the supplementary material *Jupyter Notebook 2. ASMG Framework*, the methods shown in Chapter 3 was not able to generate usable gallbladder 3D mesh, likely because of the low data quality. However, this chapter will continue to discuss the evaluation metrics that were chosen for determining the effectiveness of the proposed framework. We will compare the meshes generated from the VGAE (not structurally normalized but usable mesh) and the meshes reconstructed from the whole modified ASMG framework.

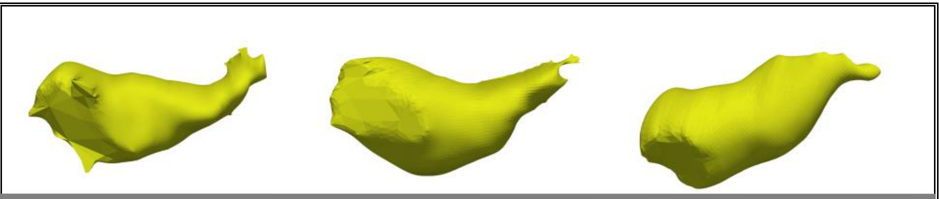
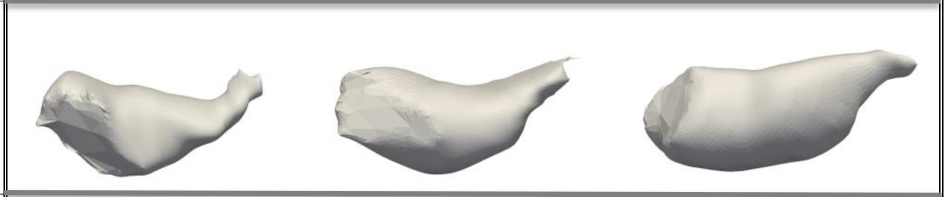
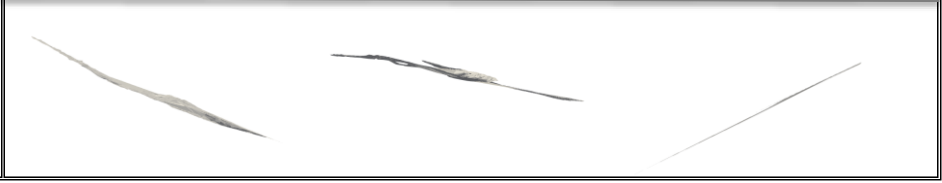
Input	
VGAE	
ASMG (modified)	

Figure 4: Examples show the comparison of different meshes. The yellow-coloured meshes in the first row represent input g_k , the second row of meshes are generated from the VGAE Φ while the last row of (needle-looking) meshes are the structurally normalized shapes generated from the modified ASMG framework.

4.3.1 Generation Quality

The experiments carried out focus on evaluating the generation quality of the framework. The metrics chosen are Hausdorff Distance (HD) [31] and the average of minimum Euclidean (ED) distance. These are common measurements used to evaluate the generalization or specificity [24] of a generative model. Generalization measures the model's capacity when dealing with data not included in the training set, which will be reflected by allowing the model to reconstruct unseen 3D gallbladder mesh. Specificity on the other hand measures the model's ability to differentiate the objects shown based on their validity by making comparisons with the training set samples.

Table 4.3.1: *Generation Quality*: comparison of meshes generated from VGAE Φ and from the modified ASMG framework (shown in bold), using the metrics HD and ED.

	VGAE Φ	**ASMG (modified)**
HD	$8.20e+1$	$\mathbf{1.29e+9}$
ED	$5.76e+1$	$\mathbf{4.48e+8}$

As demonstrated in Table 4.3.1 above, the meshes produced from the modified ASMG framework show results that leave a lot to be desired. The meshes produced from the VGAE Φ, which is included for reference instead of actual comparison, show a more reasonable value range, as the meshes used for evaluation are of more reasonable quality. If the modified ASMG proposed in this project were refined in the future, which will be discussed in detail in the following Chapter (especially Section 5.1.1 – 5.1.2), the generated meshes should produce values somewhere in between the two approaches shown in the table for both metrics.

Chapter 5 Conclusion

In this study, we attempted to apply a modified version of the ASMG framework proposed by Kalaie *et al* [21] to the AMOS dataset [30], specifically the gallbladder organ. Despite completing the implementation of the end-to-end pipeline and having reasonable I/O in each of the modules in the framework, the meshes produced in the end did not meet our expectations. Future enhancements that are suggested through careful revision of the implementation will be discussed in detail in the following section.

5.1 Future Work

5.1.1 Refinement Loss

The original ASMG framework presented by Kalaie [21] mentions a Refinement Loss which is a combination of Chamfer and Laplacian losses. The former measures the node distances between the original graph g_k and the domain transformed graph $g_k^{'}$. The latter is a Laplacian smoothness loss that results in a smoother reconstructed surface. However, there is limited documentation on the implementation of both calculations. Access to the source code can provide an understanding of the details of adapting Refinement Loss to this project.

5.1.2 Better Data Quality

To begin with enhancing the data quality, a better segmentation method can be applied on the data. This project makes use of the labels provided along with the dataset and there are misclassifications for the labels provided, leaving room for improvement for the targeted organ image extracted. There are cases where the labelled prostate goes up to the lungs, or the labelled gallbladder has the shape of a marble. Also, there can be a process of inserting interpolated slices in between the NIFTI images extracted to generate smoother meshes.

5.1.3 Other organs and generation frameworks

Future research may see the experiment of applying the same framework against other organs. Other generation model, such as Generative Adversary Network (GAN) [25] can also be swapped out for the ß-VAE in this project in future experiments.

5.1.4 Hyperparameter tuning and early stopping

In terms of optimizing the models deployed, hyperparameter tuning can be implemented if more computational power is acquired. The aim will be to find the optimum hyperparameter combination for the two models in the framework, the VGAE as well as the β-VAE. Other optimization techniques like early stopping can be beneficial in the optimization stage as well.

5.1.5 Evaluation

Future research can also perform a more thorough experiment by evaluating the matching quality of the ASM framework, as seen in the original ASM setup [21].

Appendix A Dataset Information

[1] Dataset download link:
https://zenodo.org/record/7155725#.Y0OOCOxBztM.

[2] Organ categories in the dataset:

1. Spleen
2. Right kidney
3. Left kidney
4. Gallbladder
5. Esophagus
6. Liver
7. Stomach
8. Aorta
9. Interior vena cava
10. Pancreas
11. Right adrenal gland
12. Left adrenal gland
13. Duodenum
14. Bladder
15. Prostate/uterus

Appendix B Model Structures

Table 1: VGAE Φ Structure. $|V|$ represents the cardinality of input shape.

#	Layer	Input size	Output size
1	FeaStConv	$|V| \times 3$	$|V| \times 64$
2	Batch Norm	$|V| \times 64$	$|V| \times 64$
3	FeaStConv	$|V| \times 64$	$|V| \times 64$
4	Batch Norm	$|V| \times 64$	$|V| \times 64$
5	FeaStConv	$|V| \times 64$	$|V| \times 128$
6	Batch Norm	$|V| \times 128$	$|V| \times 128$
7	FeaStConv (μ)	$|V| \times 128$	$|V| \times 128$
8	FeaStConv (log σ)	$|V| \times 128$	$|V| \times 128$
9	FeaStConv	$|V| \times 128$	$|V| \times 128$
10	Batch Norm	$|V| \times 128$	$|V| \times 128$
11	FeaStConv	$|V| \times 128$	$|V| \times 64$
12	Batch Norm	$|V| \times 64$	$|V| \times 64$
13	FeaStConv	$|V| \times 64$	$|V| \times 64$
14	Batch Norm	$|V| \times 64$	$|V| \times 64$
15	FeaStConv	$|V| \times 64$	$|V| \times 3$

Table 2: β-VAE Structure. $|V|$ represents the cardinality of input shape.

#	Layer	Input size	Output size
1	Linear	$|V| \times 3$	$|V| \times 512$
2	Batch Norm	$|V| \times 512$	$|V| \times 512$
3	Linear	$|V| \times 512$	$|V| \times 256$
4	Batch Norm	$|V| \times 256$	$|V| \times 256$
5	Linear	$|V| \times 256$	$|V| \times 768$
6	Batch Norm	$|V| \times 768$	$|V| \times 768$
7	Linear	$|V| \times 768$	$|V| \times 128$
8	Batch Norm	$|V| \times 128$	$|V| \times 128$
9	Linear	$|V| \times 128$	$|V| \times 64$
10	Batch Norm	$|V| \times 64$	$|V| \times 64$
11	Linear (μ)	$|V| \times 64$	$|V| \times 64$
12	Linear (log σ)	$|V| \times 64$	$|V| \times 64$
13	Linear	$|V| \times 64$	$|V| \times 128$
14	Batch Norm	$|V| \times 128$	$|V| \times 128$
15	Linear	$|V| \times 128$	$|V| \times 768$
16	Batch Norm	$|V| \times 768$	$|V| \times 768$
17	Linear	$|V| \times 768$	$|V| \times 256$
18	Batch Norm	$|V| \times 256$	$|V| \times 256$
19	Linear	$|V| \times 256$	$|V| \times 512$
20	Batch Norm	$|V| \times 512$	$|V| \times 512$
21	Linear	$|V| \times 512$	$|V| \times 3$

Bibliography

[1] Sidey-Gibbons JA, Sidey-Gibbons CJ. Machine learning in medicine: a practical introduction. BMC medical research methodology. 2019 Dec;19:1-8.

[2] Keshta I, Odeh A. Security and privacy of electronic health records: Concerns and challenges. Egyptian Informatics Journal. 2021 Jul 1;22(2):177-83.

[3] Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems. 2020 Mar 24;32(1):4-24.

[4] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. IEEE transactions on neural networks. 2008 Dec 9;20(1):61-80.

[5] Shuman DI, Narang SK, Frossard P, Ortega A, Vandergheynst P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE signal processing magazine. 2013 Apr 5;30(3):83-98.

[6] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems. 2016;29.

[7] Chen J, Ma T, Xiao C. Fastgcn: fast learning with graph convolutional networks via importance sampling. arXiv preprint arXiv:1801.10247. 2018 Jan 30.

[8] Wu F, Souza A, Zhang T, Fifty C, Yu T, Weinberger K. Simplifying graph convolutional networks. International conference on machine learning 2019 May 24 (pp. 6861-6871). PMLR.

[9] Levie R, Monti F, Bresson X, Bronstein MM. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. IEEE Transactions on Signal Processing. 2018 Nov 4;67(1):97-109.

[10] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. Advances in neural information processing systems. 2017;30.

[11] Monti F, Boscaini D, Masci J, Rodola E, Svoboda J, Bronstein MM. Geometric deep learning on graphs and manifolds using mixture model cnns. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 5115-5124).

[12] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. arXiv preprint arXiv:1710.10903. 2017 Oct 30.

[13] Kipf TN, Welling M. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308. 2016 Nov 21.

[14] Kingma DP, Welling M. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114. 2013 Dec 20.

[15] Litany O, Bronstein A, Bronstein M, Makadia A. Deformable shape completion with graph convolutional autoencoders. InProceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 1886-1895).

[16] Ranjan A, Bolkart T, Sanyal S, Black MJ. Generating 3D faces using convolutional mesh autoencoders. InProceedings of the European conference on computer vision (ECCV) 2018 (pp. 704-720).

[17] Verma N, Boyer E, Verbeek J. Feastnet: Feature-steered graph convolutions for 3d shape analysis. InProceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 2598-2606).

[18] Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, Lerchner A. beta-vae: Learning basic visual concepts with a constrained variational framework. InInternational conference on learning representations 2016 Nov 4.

[19] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence. 2013 Mar 7;35(8):1798-828.

[20] Beetz M, Banerjee A, Grau V. Generating subpopulation-specific biventricular anatomy models using conditional point cloud variational autoencoders. InInternational Workshop on Statistical Atlases and Computational Models of the Heart 2021 Sep 27 (pp. 75-83). Cham: Springer International Publishing.

[21] Kalaie S, Bulpitt AJ, Frangi AF, Gooya A. A Geometric Deep Learning Framework for Generation of Virtual Left Ventricles as Graphs. InMedical Imaging with Deep Learning 2023 Apr 4.

[22] Ji Y, Bai H, Ge C, Yang J, Zhu Y, Zhang R, Li Z, Zhanng L, Ma W, Wan X, Luo P. Amos: A large-scale abdominal multi-organ benchmark for versatile medical image segmentation. Advances in Neural Information Processing Systems. 2022 Dec 6;35:36722-32.

[23] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014 Dec 22.

[24] Styner MA, Rajamani KT, Nolte LP, Zsemlye G, Székely G, Taylor CJ, et al. Evaluation of 3D Correspondence Methods for Model Building. Lecture Notes in Computer Science. 2003;2732:63–75.

[25] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. Advances in neural information processing systems. 2014;27.

[26] Schlager S, Zheng G, Li S, Szekely G. Statistical shape and deformation analysis. InMorpho and Rvcg–Shape Analysis in R: R-packages for geometric morphometrics, shape analysis and surface manipulations. 2017 (pp. 217-56). Academic press.

[27] Gooya A, Lekadir K, Castro-Mateos I, Pozo JM, Frangi AF. Mixture of probabilistic principal component analyzers for shapes from point sets. IEEE transactions on pattern analysis and machine intelligence. 2017 May 2;40(4):891-904.

[28] Cosentino F, Raffa GM, Gentile G, Agnese V, Bellavia D, Pilato M, Pasta S. Statistical shape analysis of ascending thoracic aortic aneurysm: correlation between shape and biomechanical descriptors. Journal of Personalized Medicine. 2020 Apr 22;10(2):28.

[29] Harshvardhan GM, Gourisaria MK, Pandey M, Rautaray SS. A comprehensive survey and analysis of generative models in machine learning. Computer Science Review. 2020 Nov 1;38:100285.

[30] Multi-Modality Abdominal Multi-Organ Segmentation Challenge 2022 - Grand Challenge [Internet]. grand-challenge.org. [cited 2023 Sep 3]. Available from: https://amos22.grand-challenge.org/

[31] Taha AA, Hanbury A. An efficient algorithm for calculating the exact Hausdorff distance. IEEE transactions on pattern analysis and machine intelligence. 2015 Mar 3;37(11):2153-63.