GENERATION OF PATIENT SPECIFIC CARDIAC CHAMBER MODELS USING GENERATIVE NEURAL NETWORKS UNDER A BAYESIAN FRAMEWORK FOR ELECTROANATOMICAL MAPPING

Sunil Mathew [1*], **Jasbir Sra** [2], and **Daniel B. Rowe** [1]

[1]Department of Mathematical and Statistical Sciences, Marquette University, Milwaukee, WI-53233
[2]Electrophysiology, Aurora Healthcare, Milwaukee, WI-53215

ABSTRACT

Electroanatomical mapping is a technique used in cardiology to create a detailed 3D map of the electrical activity in the heart. It is useful for diagnosis, treatment planning and real time guidance in cardiac ablation procedures to treat arrhythmias like atrial fibrillation. A probabilistic machine learning model trained on a library of CT/MRI scans of the heart can be used during electroanatomical mapping to generate a patient-specific 3D model of the chamber being mapped. The use of probabilistic machine learning models under a Bayesian framework provides a way to quantify uncertainty in results and provide a natural framework of interpretability of the model. Here we introduce a Bayesian approach to surface reconstruction of cardiac chamber models from a sparse 3D point cloud data acquired during electroanatomical mapping. We show how probabilistic graphical models trained on segmented CT/MRI data can be used to generate cardiac chamber models from few acquired locations thereby reducing procedure time and x-ray exposure. We show how they provide insight into what the neural network learns from the segmented CT/MRI images used to train the network, which provides explainability to the resulting cardiac chamber models generated by the model.

1 INTRODUCTION

Electroanatomic mapping (EAM) is part of Electrophysiology studies (EPS) which are minimally invasive procedures that are useful for diagnosis, treatment planning and real time guidance in cardiac ablation procedures to treat arrhythmias like atrial fibrillation which affects millions of people in the United States. It is estimated that over 12 million people in the United States will be affected by atrial fibrillation by 2030 Colilla et al. (2013). It is characterized by rapid and irregular beating of the atria (upper chambers of the heart) which can lead to blood clots, stroke, heart failure and other heart-related complications. Cardiac ablation is a procedure used to treat atrial fibrillation by eliminating the heart tissue that causes the abnormal heart rhythm. EAM is performed by an electrophysiologist prior to ablation to create a 3D map of the heart's electrical activity to identify the location of the abnormal heart tissue. The electrophysiologist uses this map to guide the catheter to the location of the abnormal heart tissue and eliminate it using radiofrequency (RF) energy or cryogenic energy.

During an RF ablation procedure, the electrode at the tip of the catheter is placed in contact with the tissue causing the arrhythmia, and high-frequency electrical energy is delivered to create a localized lesion or scar. The goal of RF ablation is to disrupt the abnormal electrical pathways in the heart causing the arrhythmia and create a new pathway that follows the normal electrical conduction system of the heart, thus restoring normal heart rhythm and reducing or eliminating the need for medication to control the arrhythmia.

Minimizing the x-ray exposure from the fluoroscope to the patient is one of the electrophysiologist's goals while performing electroanatomical mapping. Using a probabilistic machine learning model in an electroanatomical mapping system can provide a good approximation of the mapped chamber with a few acquired locations from the mapping catheter in the chamber of interest. This considerably reduces the time taken to map the chamber, thus minimizing the x-ray exposure from the fluoroscope.

2 ELECTROANATOMICAL MAPPING SYSTEMS

An EAM system is a state-of-the-art medical device that is used for navigation during cardiac ablation procedures and to create a detailed and accurate map of the electrical activity in a patient's heart. The system typically consists of a computer, high-definition monitors that are strategically positioned in front of the electrophysiologist during the procedure, a specialized catheter with electrodes in its distal end that captures electrical information, and specialized software that is used to analyze signals and create a 3D electroanatomical map of the chamber of interest.

EAM systems vary based on the technology that is used to capture the location of the catheter. The commonly used technologies are magnetic location tracking Gepstein et al. (1997), impedance based location tracking and image processing Sra et al. (2016). EAM systems that are based on magnetic location tracking places a device under the patient table to generate a magnetic flux which is used to locate the catheter. EAM systems that use impedance based tracking utilizes body surface patches to measure the impedance and locate the catheter. EAMs that use image processing techniques tracks the catheter in fluoroscope images to converts them to 3D location.

The high accuracy and precision of EAM systems have revolutionized the diagnosis and treatment of various cardiac arrhythmias. However the maps produced using modern electranatomical systems do not produce anatomically accurate chamber models when only few points are acquired. The shape produced is representative of the points acquired and does not use any

*correspondence: sunil.mathew@marquette.edu

Preprint – Generation of patient specific cardiac chamber models using generative neural networks under a Bayesian framework for electroanatomical mapping

2

prior information pertaining to the shape of the chamber being mapped and may not produce an accurate anatomical model Mukherjee et al. (2014). They only start resembling the chamber of interest when a large number of points are acquired covering all areas of the chamber. This is a major drawback of the current electroanatomical mapping systems. We propose a surface reconstruction method that uses probabilistic machine learning models to generate a detailed anatomical model of the chamber of interest from a few points acquired by the mapping catheter. This will help in reducing the time taken to map the chamber and thus reduce the x-ray exposure from the fluoroscope.

3 Method

There are four main steps involved in our approach to generate a 3D cardiac chamber model from 3D location and electrical information acquired by an electroanatomical mapping system. The block diagram shown in figure 1 visualizes the information flow in the system. The data collected by the 3D mapping system is passed on to the Data processing step, then to the trained Probabilistic Machine Learning Model, and the output of the machine learning model is fed into the surface reconstruction module in the last step to generate a patient specific cardiac model. Each block is explained in detail in following sections.

Figure 1: Generation of 3D cardiac chamber models

3.1 Data processing

The points acquired during electroanatomical mapping are in the coordinate space of the mapping system. The 3D location information acquired during mapping is in a raw unstructured format. The data is preprocessed to transform it into voxel data Xu2 (2021) that can be fed into the machine learning model. Voxels are the 3D equivalent of pixels in 2D images. They are the smallest unit of a 3D image. The 3D location data is transformed into voxel space using the following equation

$$v = \frac{p - p_{min}}{p_{max} - p_{min}} * n$$

where p is the 3D location of the point, p_{min} and p_{max} are the minimum and maximum 3D location values defined by the field-of-view (FOV), n is the number of voxels in each dimension and v is the voxel location. The number of voxels in each dimension is chosen based on the number of voxels in each dimension of the training data used to train the machine learning model. Mapping systems tend to match the FOV of the fluoroscope to project 3D models onto the 2D fluoroscope image.

The next step is compute the convex hull of all acquired points. A Delaunay 3D triangulation algorithm with an alpha value which produces a closed surface can be used to get an alpha shape Edelsbrunner and Mücke (1994). The alpha value is the radius of a ball that determines whether a simplex (vertex, edge,

face or tetrahedron) in the Delaunay triangulation is part of the alpha shape or not. An alpha value of zero results in a convex hull. The resulting simplices of the Delaunay triangulation can be used to estimate which voxels are inside versus outside the alpha shape. The voxels inside are set to 1 and the voxels outside are set to zero. Different values of alpha can be chosen to obtain a concave hull that might produce different results, but they were not explored as they may cause the surface to have holes with higher values of alpha. The resulting voxels are then passed onto the machine learning model to generate inference.

3.2 Probabilistic Machine Learning Model

Probabilistic Machine Learning models are models that use probability theory and statistics to represent uncertainty and variability in data and the learning processes. They can be used for tasks such as classification, regression, clustering, anomaly detection, generative modeling and reinforcement learning. Unlike regular neural networks they are more interpretable. Here they are used in the context of generative modeling.

Unlike a discriminative model which attempts to learn a boundary between different kinds of data points, generative models aim to learn the underlying structure and distribution of the data, and then generate new data that resembles the original data. They can capture complex and high-dimensional patterns in the data that are difficult to model explicitly. The goal here is to learn a representation of the chamber of interest (left atrium of the heart). We will be using latent variable models for this purpose. At the core of generative modeling is Bayesian inference, which uses Baye's rule to obtain the posterior distribution of the parameters of the data model,

$$P(\theta|D) = \frac{P(D|\theta).P(\theta)}{P(D)}$$

where $P(D|\theta)$ is the data likelihood with parameters θ of the model, $P(\theta)$ is the prior probability, and $P(D)$ is the evidence which can be computed as sum over all possible values of θ. This distribution is then utilized to generate inference that can provide insight into the learning process and the model. When there is a large number of parameters in the model which is the case in neural networks, the denominator term in (3.2) becomes intractable.

This marginal likelihood is calculated using the sum and product rule of probability. This denominator term known as evidence is a normalizing constant which can be omitted to get the relationship,

$$P(\theta| D) \propto P(D|\theta)P(\theta).$$

These formulas above form the core of Bayesian inference. The posterior distribution is proportional to the product of the prior and the likelihood. The prior distribution is the distribution of the parameters before seeing the data. The likelihood is the probability of the data given the parameters. The posterior distribution is the distribution of the parameters after seeing the data.

PREPRINT – GENERATION OF PATIENT SPECIFIC CARDIAC CHAMBER MODELS USING GENERATIVE NEURAL NETWORKS UNDER A BAYESIAN FRAMEWORK FOR ELECTROANATOMICAL MAPPING

3

In a generative model where latent variables z are a subset of parameters θ. The following equation represents the generative model, where $\tilde{x}$ is the generated data, $\mathbf{f}$ is a function that maps a latent variable $\mathbf{z}$ to the generated data.

$$\tilde{x} = \mathbf{f(z)}$$

The prior distribution of the latent variable $\mathbf{z}$ is represented by the equation below.

$$\mathbf{z} \sim p(\mathbf{z})$$

The prior distribution is used to sample from the latent space to generate new data. A new data point $\mathbf{x}$ can be used to update the prior distribution of the latent variable $\mathbf{z}$ using Bayes' rule. The posterior distribution is given by,

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}$$

where $p(\mathbf{x} \mid \mathbf{z})$ is the likelihood. This posterior distribution is intractable due the high dimensional integral posed by the large number of parameters in the neural network. To address this problem we employ approximate Bayesian inference techniques. The two main techniques of approximate Bayesian inference are Markov chain Monte Carlo (MCMC) methods and Variational Bayes, also known as Variational Inference (VI). In the following sections we describe how segmented CT/MRI images can be modeled using approximate Bayesian inference techniques and how the results compare in the context of generating patient specific cardiac models. First we present the data and then the two different approximate Bayesian inference techniques that can be employed to generate patient specific cardiac models.

3.2.1 Training Data

In probabilistic machine learning, training data is essential for building and fine-tuning models. The quality and quantity of training data directly impact the performance of the model, including its accuracy, precision, and generalizability. These models use training data to learn patterns and relationships between the input features and the output variable.

Training data that is biased, incomplete, or inaccurate can negatively impact the model's performance. Therefore, it's essential to carefully select and preprocess the training data to ensure its quality and relevance to the problem at hand. Left atrial anatomy can be highly variable. Its important that every data point in the dataset has the same amount of pulmonary vein segmented along with the left atrial cavity. The datasets also needs to be normalized as some scans may have a different field of view or voxel size. Voxels belonging to both the left atrial cavity and the myocardium need to be segmented together as one class.

The training data used to train the PML model consists of binary voxels with zeros for the background and ones for the left atrium and pulmonary veins. The individual slices stack up to become a $n_x \times n_y \times n_z$ voxel grid to form one 3D data point.
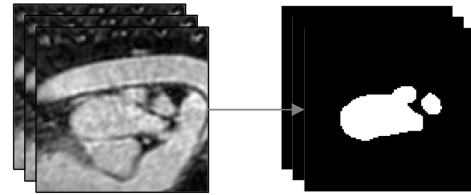

Figure 2: Segmented Left Atrial data

3.2.2 Restricted Boltzmann Machine Model for Segmented Left Atrial Data

Restricted Boltzmann Machines (RBM) Smolensky (1986); Freund and Haussler (1991); Hinton (2002) are energy based stochastic artificial neural networks that can learn a probability distribution over its inputs. The structure of an RBM for segmented left atrial data is shown below in Figure 3. It consist of
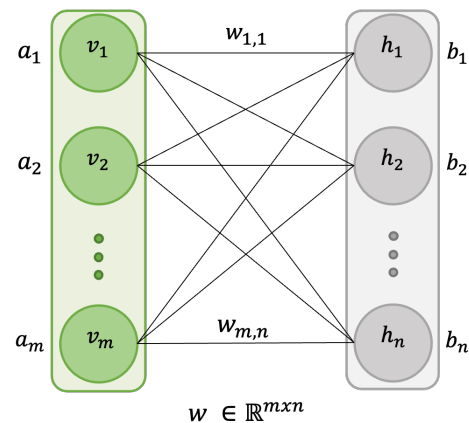

Figure 3: Structure of a Restricted Boltzmann machine

binary visible (v) and hidden (h) neurons connected by a $m \times n$ weight matrix $W_{m \times n}$. There are no interconnections between the visible nodes or the hidden nodes and they form a bipartite graph. There is a vector of bias a_m associated with the visible nodes and another vector of bias b_m associated with the hidden nodes. The energy of a pair of v, h is given by the following,

$$E(v, h) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}v_i h_j$$

$$E(v, h) = -a^T v - b^T h - v^T W h.$$

Here the visible nodes represent the voxels of a segmented CT/MRI data and the hidden nodes represent features learned from the dataset. There can be three kinds of parameters introduced into this network. The parameter w_{ij} for strength between a visible and hidden node, b_i for every visible node, c_j for every hidden node. The joint probability distribution of this MRF specifies the probability the network assigns to a pair of visible and hidden vector and can be written as,

$$P(v, h) = \frac{1}{Z}e^{-E(v, h)}$$

where the partition function sums over all such pairs of v, h,

$$Z = \sum_{V}\sum_{H} e^{-E(v, h)}$$

Preprint – Generation of patient specific cardiac chamber models using generative neural networks under a Bayesian framework for electroanatomical mapping

4

and is the normalizing constant to make the probability distribution sum to 1. The probability for an image data or visible vector can be obtained by summing over all the possible hidden vector configurations

$$P(v) = \frac{1}{Z} \sum_H e^{-E(v,\, h)}.$$

When there are more than a few hidden units, it becomes difficult to compute the partition function as it will have exponentially many terms. So Markov chain Monte Carlo methods such as Gibbs sampling is used to obtain samples from the model starting from a global configuration. The conditional distributions for the hidden and visible units are used to construct a Markov chain and run until it reaches its stationary distribution. The probability of a global configuration at thermal equilibrium is an exponential function of its energy. Since there are no connection between nodes in a group they are independent. The conditional probability for visible nodes given hidden nodes are

$$P(v|h) = \prod_i^m P(v_i|h).$$

This probability values allow us to know how much of the current configuration of hidden nodes are represented by the image and conversely the conditional probability for hidden nodes given the visible nodes are

$$P(h|v) = \prod_j^n P(h_j|v).$$

From the above probabilities we can also estimate the probability of one of the visible or hidden node being activated. Since there is no connection between visible nodes,

$$P(h_j = 1|v) = \sigma(\sum_{i=1}^m w_{ij} v_i + b_j)$$

$$P(h|v) = \prod_{j=1}^n \sigma(v^T W_{:j} + b_j) = \sigma(v^T W_{:i} + b_j)$$

where σ denotes the logistic sigmoid. Similarly,

$$P(v_i = 1|h) = \sigma(\sum_{j=1}^n w_{ij} h_j + a_i)$$

$$P(v|h) = \prod_{i=1}^m \sigma(v^T W_{:i} + a_i).$$

3.2.3 Training the RBM Model

The goal of training a RBM to be a generative model is to maximize the likelihood of the training dataset of images. This may prove to be difficult but Hinton (2002) showed that there is a less obvious objective function than log likelihood of the data to optimize called the Contrastive Divergence which is the difference between two KL divergences. It consists of performing block Gibbs sampling and batch updates to the weights and biases of the network similar to stochastic gradient descent in a regular neural network that uses back propagation for training. The weights and biases of the network are adjusted to lower the
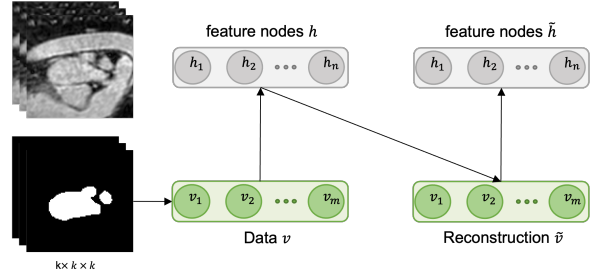

Figure 4: Single training step of the model

energy of the training image and raise the energy of other images which are low in energy, thereby making a large contribution to the partition function.

The gradient of the log-likelihood with respect to the weights is given by

$$\frac{\partial}{\partial w_{ij}} \log P(v) = \langle h_i v_j \rangle_{\text{data}} - \langle h_i v_j \rangle_{\text{recon}}$$

where $\langle . \rangle_{\text{data}}$ and $\langle . \rangle_{\text{recon}}$ denote the expectation over the data and the model distributions, respectively. The second term in this equation is difficult to compute.

Algorithm 1 k-Contrastive Divergence algorithm

Require: $\mathbf{v}$: visible units, $\mathbf{W}$: weights, $\mathbf{b}$: visible bias, $\mathbf{c}$: hidden bias, k: number of Gibbs sampling steps, α: learning rate
1: **while** not converged **do**
2:     **for** $i = 1$ to $n_{batches}$ **do**   ▷ For each batch (v) of images
3:         $\mathbf{h} \leftarrow \sigma(\mathbf{W} \cdot \mathbf{v}^{(i)} + \mathbf{c})$ ▷ Positive phase
4:         $\mathbf{v}_0 \leftarrow \mathbf{v}^{(i)}$
5:         **for** $j = 1$ to k **do**   ▷ Negative phase
6:             $\mathbf{v}_j \leftarrow \sigma(\mathbf{W}^T \cdot \mathbf{h} + \mathbf{b})$
7:             $\mathbf{h} \leftarrow \sigma(\mathbf{W} \cdot \mathbf{v}_j + \mathbf{c})$
8:         **end for**
9:         $\mathbf{W} \leftarrow \mathbf{W} + \alpha \cdot (\mathbf{v}_0 \cdot \mathbf{h}^T - \mathbf{v}_k \cdot \mathbf{h}_k^T)$
10:        $\mathbf{b} \leftarrow \mathbf{b} + \alpha \cdot (\mathbf{v}_0 - \mathbf{v}_k)$
11:        $\mathbf{c} \leftarrow \mathbf{c} + \alpha \cdot (\mathbf{h}^{(0)} - \mathbf{h}_k)$
12:     **end for**
13: **end while**

In contrastive divergence learning instead of starting with a randomly initialized visible vector and running the Gibbs chain for a large number of steps, a data point from the training set is used as the initial value for the visible vector and k steps of Gibbs sampling is used to get a reconstruction ($k = 1$ works well) to get the second term from what is known as the negative phase of contrastive divergence learning. The difference between the positive and negative phase is used to update the weights and biases. The algorithm is summarized in Algorithm 1. The update rule seen in step (9) in the algorithm above can be written as,

$$\Delta w_{ij} = \epsilon \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon} \right)$$

where ϵ is the learning rate, w_{ij} is the weight connecting the visible unit i and hidden unit j, $\langle v_i h_j \rangle_{data}$ is the expected value

of the product of the visible unit i and hidden unit j under the data distribution, and $\langle v_i h_j \rangle_{recon}$ is the expected value of the product of the visible unit i and hidden unit j under the model distribution. The update rule for the biases are given by,

$$\Delta b_i = \epsilon \left(\langle v_i \rangle_{data} - \langle v_i \rangle_{recon} \right)$$

$$\Delta c_j = \epsilon \left(\langle h_j \rangle_{data} - \langle h_j \rangle_{recon} \right)$$

where $\langle v_i \rangle_{data}$ is the expected value of the visible unit i under the data distribution, and $\langle v_i \rangle_{recon}$ is the expected value of the visible unit i under the model distribution. The algorithm is run until the weights and biases converge to produce accurate reconstructions of the segmented data. The weights and biases represent what the model has learned from the training data and is utilized for inference.

3.2.4 Model Interpretability and Inference

After the training procedure, it is possible to examine each feature and infer what the model has learned from the training dataset. As each voxel is connected to all nodes in the feature vector, so we can represent the weights as a 3D voxel grid. Figure 5 shows weights of one of the hidden nodes represented as a voxel grid, before and after pruning low magnitude weights. The voxels in the cavity of the left atrium are of higher magnitude suggesting that the model has more certainty on classifying them as being part of the foreground or the left atrium.
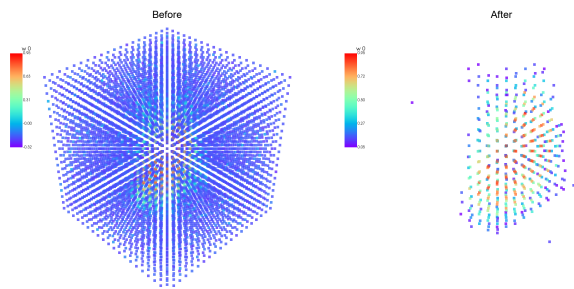

Figure 5: 3D RBM weights before(left) and after(right) pruning.

Different nodes within the network learn distinct features, and there may be correlations among these features. When there are only few hidden nodes, each node has a more significant impact on the representation of the volume data. The hidden nodes are forced to work together and correlate their activity to reconstruct the volume data effectively. Consequently, the features learned by the RBM tend to have higher interdependence and can exhibit strong correlations. However, as the number of hidden nodes increases, each hidden node has more freedom to capture different aspects of the data independently. With more nodes, the hidden layer can distribute the workload among them, allowing different nodes to specialize in capturing different aspects of the data. This can lead to a reduction in the correlation between features and improve quality of reconstruction.

After training the RBM, we can obtain the posterior predictive distribution by sampling from the posterior distribution of the hidden nodes.

The posterior probabilities of hidden nodes given visible nodes, trained weights, hidden biases are calculated as

$$p(h_j = 1|v, W, c) = \sigma \left(\sum_i W_{ij} v_i + c_j \right)$$

where h_j is the j-th hidden unit.

A binary sample from the posterior distribution of hidden nodes are drawn as

$$h_{s_{ij}} \sim \text{Bernoulli} \left(p(h_j = 1|v, W, c) \right)$$

where $h_{s_{ij}}$ is the i-th sample of the j-th hidden unit.

A sample of a hidden vector is drawn as

$$h_s \sim \text{Bernoulli}(p(h|v, W, c))$$

The posterior probabilities of visible nodes given hidden nodes are calculated as

$$p(v_i = 1|h_s, W, b) = \sigma \left(\sum_j W_{ij} h_{s_{ij}} + b_i \right).$$

The mean and standard deviation of the posterior predictive distribution are calculated as

$$v_{mean} = \frac{1}{n} \sum_{s=1}^{n} p(v|h_s, W, b)$$

$$v_{std} = \frac{1}{n} \sqrt{\sum_{s=1}^{n} (p(v|h_s, W, b) - v_{mean})^2}.$$

where n is the number of samples drawn from the posterior distribution of the hidden nodes. The mean and standard deviation of the posterior predictive distribution are used to reconstruct the volume and quantify the uncertainty in the reconstruction.
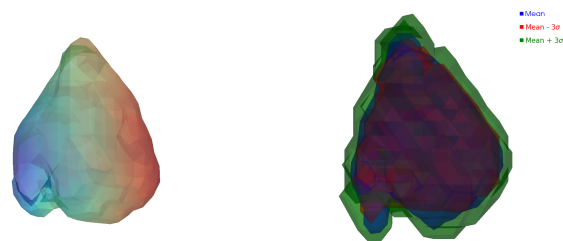

Figure 6: True surface (left)mean(blue), mean+std (green), mean-std (red) surfaces from RBM model output

Figure 6 shows the mean, mean +/- standard deviation surfaces reconstructed using posterior probability distribution of the hidden nodes of an RBM.

3.2.5 A Variational Autoencoder (VAE) model for volumetric data

MCMC methods allow for the computation of posterior distributions with high accuracy, as they asymptotically converge to

PREPRINT – GENERATION OF PATIENT SPECIFIC CARDIAC CHAMBER MODELS USING GENERATIVE NEURAL NETWORKS UNDER A BAYESIAN FRAMEWORK FOR ELECTROANATOMICAL MAPPING

6

the true posterior distribution. They are able to sample from complex, high dimensional distributions with multiple modes and non-convex shapes. However, they come with significant computational costs, that is where Variational Inference (VI) techniques can prove to be useful. Variational Inference Peterson and Anderson (1987); Jordan et al. (1999) aims at posing the intractable inference problem as one of optimization. It approximates the posterior distribution of a latent variable model by an easy to sample distribution such as a Gaussian distribution.

A Variational Autoencoder (VAE) Kingma and Welling (2014); Rezende et al. (2014) is a generative model that learns a compressed latent representation of the data by mapping it to a latent space defined by a prior probability distribution. It consists of an encoder network that maps the data to latent variables, and a decoder network that reconstructs the data from the latent variables. The main difference between a VAE and a standard autoencoder Hinton and Salakhutdinov (2006) is that the latent space in a VAE is stochastic. The output of the encoder is a distribution over the latent vectors, rather than a single vector. This allows the VAE to generate new data points by sampling from the latent space.

VAEs are trained to minimize the difference between the input and the reconstructed output, and the divergence between the learned latent variable and a known prior distribution. Several methods and architectures have been developed to use a VAE for 3D data. One of the most popular methods is the 3D Convolutional VAE (3D-CVAE) Wu et al. (2016) used for modeling volumetric objects. The 3D-CVAE is similar to a traditional VAE, but uses 3D convolutions instead of 2D convolutions to process 3D data. They are good at capturing local spatial information in the image data. Figure 7 shows the block diagram of a VAE that is used to reconstruct segmented CT or MRI data.
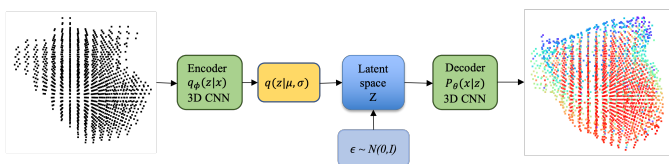

Figure 7: 3D Variational Autoencoder Model

Here we try to learn a compressed representation of segmented CT or MRI data x that is optimized to match a multivariate Gaussian distribution with mean μ and variance σ. This is achieved by introducing a latent variable z that represents the compressed representation of the volume (eg: left atrium), and training the VAE to learn a conditional probability distribution $p_\theta(x|z)$ that maps the compressed representation z to the input data x. The parameters θ of the conditional probability distribution are learned during training.

The goal is to learn a compressed representation z that maximizes the joint probability $p_\theta(x|z)p(z)$ of the data x given the compressed representation z and the prior distribution $p(z)$ over the compressed representation. Maximizing this joint probability ensures that the learned representations are both good at reconstructing the original data ($p_\theta(x|z)$ is high) and align well with our prior belief about what z should be like ($p(z)$ is high).

However, this joint distribution is intractable, as it involves integrating over all possible values of z. To overcome this problem, the VAE introduces an approximate posterior distribution $q_\phi(z|x)$ which can be used to approximate the true posterior distribution $p(z|x)$.

The VAE learns the parameters ϕ of the approximate posterior distribution $q_\phi(z|x)$ by minimizing the KL divergence between the approximate posterior and the true posterior. The KL divergence is defined as:

$$KL(q_\phi(z|x)\|p(z|x)) = \int q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z|x)} dz$$

Minimizing the KL divergence ensures that the approximate posterior distribution is as close as possible to the true posterior distribution. The loss function for the VAE is the sum of two terms: the reconstruction loss and the KL divergence loss. The reconstruction loss measures the difference between the input data and the data reconstructed from the compressed representation z using the decoder. It is defined as:

$$\mathcal{L}\text{rec}(\theta, \phi; x) = -\mathbb{E}q_\phi(z|x)[\log p_\theta(x|z)].$$

The KL divergence loss measures the difference between the approximate posterior distribution and the prior distribution:

$$\mathcal{L}\text{KL}(\phi; x) = KL(q\phi(z|x)\|p(z)).$$

The total loss function is the sum of the reconstruction loss and the KL divergence loss:

$$\mathcal{L}(\theta, \phi; x) = \mathcal{L}\text{rec}(\theta, \phi; x) + \mathcal{L}\text{KL}(\phi; x).$$

which is equivalent to maximizing the ELBO:

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)\|p(z)).$$

During training, the parameters θ and ϕ are learned by minimizing the total loss function with respect to these parameters:

$$\theta, \phi = \arg\min_{\theta,\phi} \mathcal{L}(\theta, \phi; x).$$

Training via backpropagation is made possible in VAEs using the reparameterization trick. It involves reparameterizing the latent variables z as a function of a random noise variable ϵ that can be sampled from a standard normal distribution. Specifically, we can express z as

$$z = \mu + \sigma \odot \epsilon$$

where μ and σ are the mean and standard deviation of the approximate posterior distribution $q_\phi(z|x)$, and $\odot$ denotes element-wise multiplication.

By reparameterizing z in this way, we can sample from the approximate posterior distribution $q_\phi(z|x)$ using a simple and differentiable transformation. This allows us to backpropagate

PREPRINT – GENERATION OF PATIENT SPECIFIC CARDIAC CHAMBER MODELS USING GENERATIVE NEURAL NETWORKS UNDER A BAYESIAN FRAMEWORK FOR ELECTROANATOMICAL MAPPING

7

through the sampling process, which is necessary for training the VAE using stochastic gradient descent.

This results in a model that can learn meaningful representations of the left atrium from segmented CT or MRI data, which then can be used to generate patient-specific models of the left atrium by sampling from the learned latent space when presented with seed points acquired during mapping.

We can generate a new data point x by first sampling a latent variable z from the prior distribution $p(z)$, and then passing it through the decoder to get a generated output $\tilde{x}$

$$z \sim p(z), \quad \tilde{x} = p_\theta(x|z).$$

Samples from $p_\theta(x|z)$ can be used to obtain a mean surface as well as a measure of uncertainty. The mean surface is obtained by taking the mean of the samples, and the uncertainty is obtained by taking the standard deviation of the samples

$$\tilde{x} = \mu_{z \sim p(z)}(p_\theta(x|z)), \quad \sigma_{z \sim p(z)}(p_\theta(x|z)).$$

Visualizing the Latent Space in 3D

The Figure 8 shows few representations from the 3D latent space of a VAE trained on segmented MRI images of the left atrium of the heart. The latent space of a VAE is typically high-dimensional, when a latent space is 2D or 3D, it can be visualized by sampling from a uniform 2D or 3D grid. Similarly when the latent space has a dimensionality greater than three we can use the following method to visualize the latent space from a n-dimensional uniform grid.
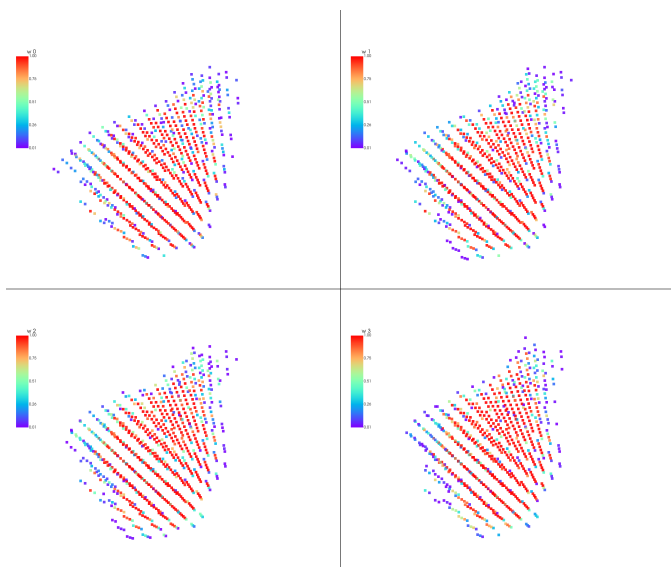

Figure 8: 3D Latent Space of a VAE trained on segmented left atrial data.

A standard normal distribution is used to sample k equally spaced elements from each dimension of the latent space defined by the n-dimensional grid.

$$d_i = q(linspace(a, b, k))$$

where $q(\cdot)$ is the quantile function of the standard normal distribution, a and b are the lower and upper bounds of each dimension of the latent space, k is the number of linearly spaced points obtained by the following equation,

$$linspace(a, b, k) = \{a, a + \frac{(b-a)}{k-1}, a + 2\frac{(b-a)}{k-1}, \ldots, b\}$$

Suppose we have an n-dimensional latent space, $D_1, D_2, \ldots, D_n$, then a Cartesian product of the n elements from each dimension can be taken to obtain all the possible ordered combinations of the n elements from each dimension.

$$\text{Cartesian Product}(D_1, D_2, \ldots, D_n) = (d_1, d_2 \ldots, d_n) \mid$$
$$d_1 \in D_1, \ldots, d_n \in D_n$$

From these ordered combinations, a linearly spaced set of points are sampled to obtain a set of points in the latent space which is then be passed through the decoder to obtain a visualization of the latent space. The latent space provides insights into the distribution of the data and the learning process. It can be used to identify outliers and anomalies in results generated by the model. This can guide the practitioner to either update the training data or update the model itself.

Surface Reconstruction

The next step in our system is Surface Reconstruction. The dense point cloud generated by the ML model is used to generate a 3D surface mesh model of the chamber of interest. The surface reconstruction algorithm used in our system is the 3D Marching Cubes algorithm Lorensen and Cline (1987), which is a popular method for generating a 3D surface mesh from volumetric data.
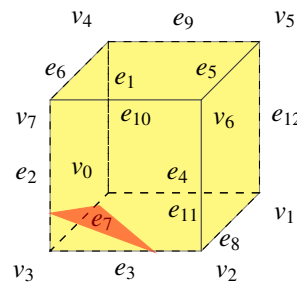

Figure 9: A voxel with an isosurface facet.

The dense 3D point cloud voxel data is divided into small cubes, each of which is then examined to determine whether it contains any parts of the left atrial surface. The algorithm considers each of the eight vertices of the cube to determine whether it lies inside or outside the surface. The scalar value at each vertex (voxel value) is compared to a threshold value. If the scalar value is greater than the threshold, the vertex is considered to be inside the surface, otherwise it is marked to be outside. The algorithm then uses the results to determine which edges of the cube are intersected by the surface and connects them to form a triangle as seen in Figure 9. The algorithm then repeats this process for each cube in the volume.

PREPRINT – GENERATION OF PATIENT SPECIFIC CARDIAC CHAMBER MODELS USING GENERATIVE NEURAL NETWORKS UNDER A BAYESIAN FRAMEWORK FOR ELECTROANATOMICAL MAPPING

8

Figure 10: 3D Surface mesh generated using marching cubes on ML output.

The resulting triangles are then connected to form a surface mesh. Figure 10 shows a 3D surface mesh generated using marching cubes on the output of the machine learning model. Here the output is the mean from the posterior distribution of the latent variable model. Other summary statistics such as variance can be used to visualize the uncertainty in the model prediction. This can be useful for guiding the electrophysiologist to regions of the chambers where more points need to acquired to get a better approximation of the chamber.

Post processing

Post processing is an important step in 3D mesh generation pipeline, as it can significantly improve the quality and usability of the mesh. Here the 3D mesh generated from the 3D marching cubes algorithm is first passed through a filter which removes small, isolated mesh components. Then it is passed through a smoothing filter and finally checked for holes and filled if any is found to get a watertight mesh that represents the left atrium.

Evaluation Metrics

The evaluation metrics used to compare the accuracy of reconstructions of the data is the dice score. The dice score measures the similarity between the ground truth and the generated data. The dice score is calculated as follows:

$$Dice = \frac{2 \times |A \cap B|}{|A| + |B|}$$

where A and B are the ground truth and the generated data, respectively. The dice score is a value between 0 and 1, where 1 is the best possible score.

4   LEFT ATRIAL DATASET

Left atrial anatomy can be highly variable across patients. While size and shape can vary, the main feature that can be used to categorize them is the number of pulmonary veins (PV). The most common type of left atrium has 2 left (LPV) and 2 right (RPV), other variants are 2 LPVs and 3 RPVs, and 1 LPV and 2 RPVs. Krum et al. (2013).

The data used in the experiments in the following sections are from Task02_Heart dataset Simpson et al. (2019) of the Medical Segmentation Decathlon held as part of the Medical Image Computing and Computer Aided Interventions (MICCAI) Conference 2018 at Granada, Spain. This dataset was originally released through the Left Atrial Segmentation Challenge (LASC) 2013. The dataset has 20 segmented MRI scans with

a voxel resolution of $1.25 \times 1.25 \times 2.7 mm^3$ covering the entire heart acquired during a single cardiac phase (free breathing with respiratory and ECG gating). The MRI scans were segmented by experts to obtain ground truth segmentation labels. All of the datasets have four pulmonary veins (74% of the population). Out of the total of 20 scans, 15 were employed for training purposes, while the remaining 5 scans were utilized for testing.

4.1   Experiments

For all experiments, the MSD dataset was cropped and down-sampled to be a $20 \times 20 \times 20$ 3D array of voxels. The system was trained for 100 epochs with a batch size of 1 for both the RBM and VAE based systems to generate inference. Point acquisition by the electroanatomical mapping system was simulated by randomly sampling points from the test data. The simulation involved generating the true surface using marching cubes, then sampling n vertices from the resulting surface mesh. The vertices are then compared to voxel locations that were used to generate the mesh by measuring their Euclidean distance. The closest voxel location is then recorded as a point acquired during mapping. Algorithm 2 summarizes the steps involved in simulating electroanatomical mapping.

Algorithm 2 EAM Simulation

Require: 3D data point (voxel data) from the test set
Require: Number of points to be acquired (n)
Ensure: Points acquired during mapping
 1: **Generate true surface:**
 2: Generate true surface s_{true} using marching cubes
 3: **Sample points from true surface s_{true}:**
 4: $v_n \leftarrow$ sample n random points from the vertices of s_{true}
 5: **for** each point p in the vertex list v_n **do**
 6:     **for** each point p_v in the voxel data **do**
 7:         Calculate Euclidean distance d between p and p_v
 8:         **if** $d <=$ threshold **then**
 9:             Add p_v to points list
10:         **end if**
11:     **end for**
12: **end for**
13: **Return the points list**

Three EP studies were simulated, the first where $n = 25$ points were acquired, the second where $n = 100$ points were acquired, and the third where $n = 100$ points were acquired. The points were then passed on to the system to generate a 3D surface mesh model of the left atrium. The 3D surface mesh model was then compared to the ground truth data using dice scores to evaluate the accuracy of the system.

5   RESULTS

The results from the three simulated experiments are discussed in this section. Table 1 shows the dice scores generated by the two systems for 20, 100, 250 points,

From Table 1, we can observe that the dice score increases as the number of points acquired increases. We can also observe that the RBM based system performs better than the VAE based system. This is because the RBM based system is able to gener-
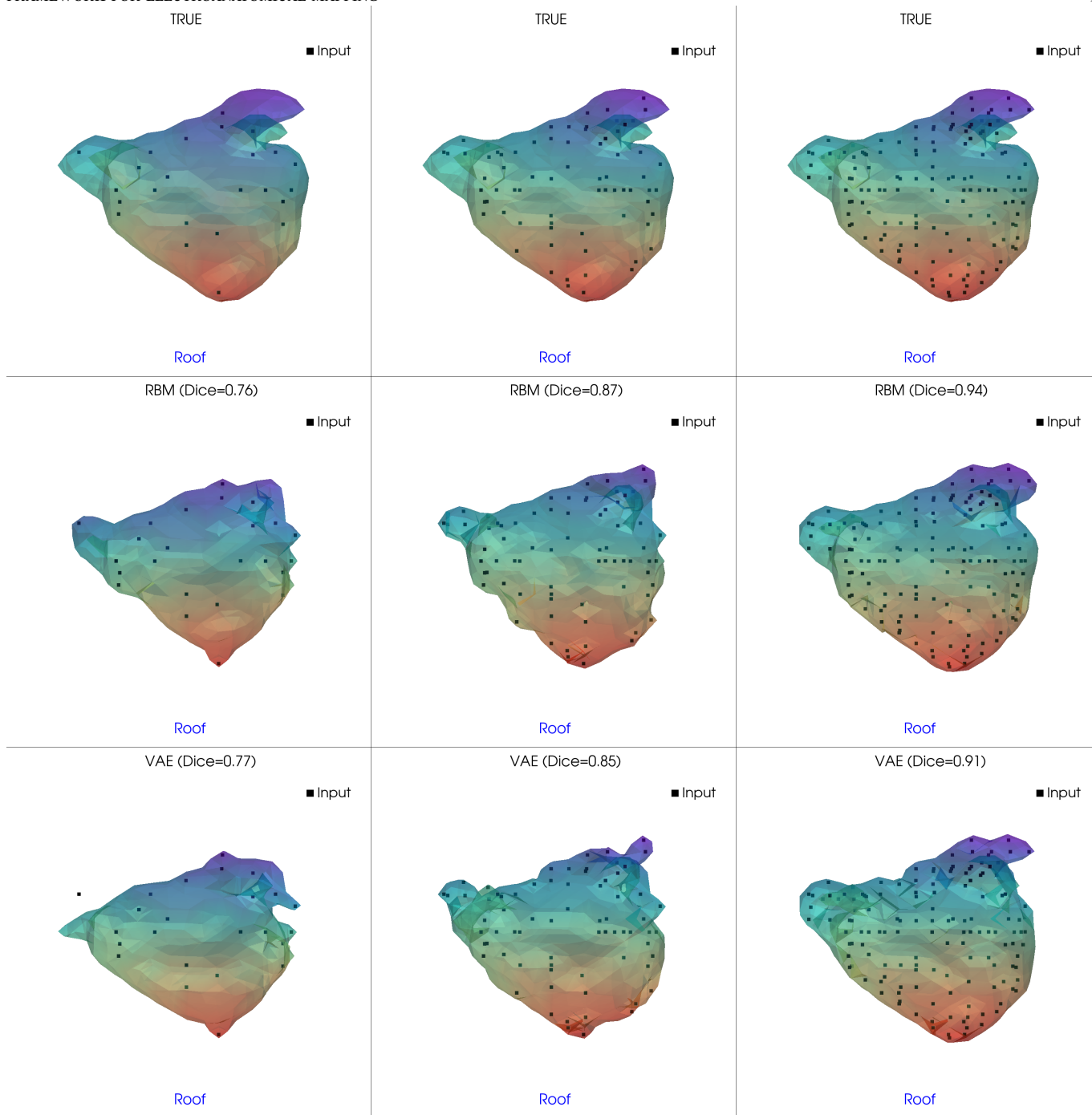
Figure 11: RBM, VAE comparison plot with 25 (left), 100 (middle) and 250 (right) points in Roof view.

Table 1: Comparison of dice scores for RBM and VAE based systems.

Model	20 points	100 points	250 points
RBM	0.79	0.89	0.92
VAE	0.76	0.83	0.88

ate a more accurate approximation of the ground truth data than the VAE based system.

Figure 11 shows comparison of surface reconstruction results using the RBM, VAE systems for a patient data from the test set in Roof view. The first row shows the surface reconstructed using the ground truth data and the points that were sampled from it for the reconstructions in the second and third row. The first column shows reconstructions using 25 points, second column shows reconstructions using 100 points, and the third column shows reconstructions using 250 points. The RBM based system is able to generate a more accurate approximation of the

PREPRINT – GENERATION OF PATIENT SPECIFIC CARDIAC CHAMBER MODELS USING GENERATIVE NEURAL NETWORKS UNDER A BAYESIAN FRAMEWORK FOR ELECTROANATOMICAL MAPPING
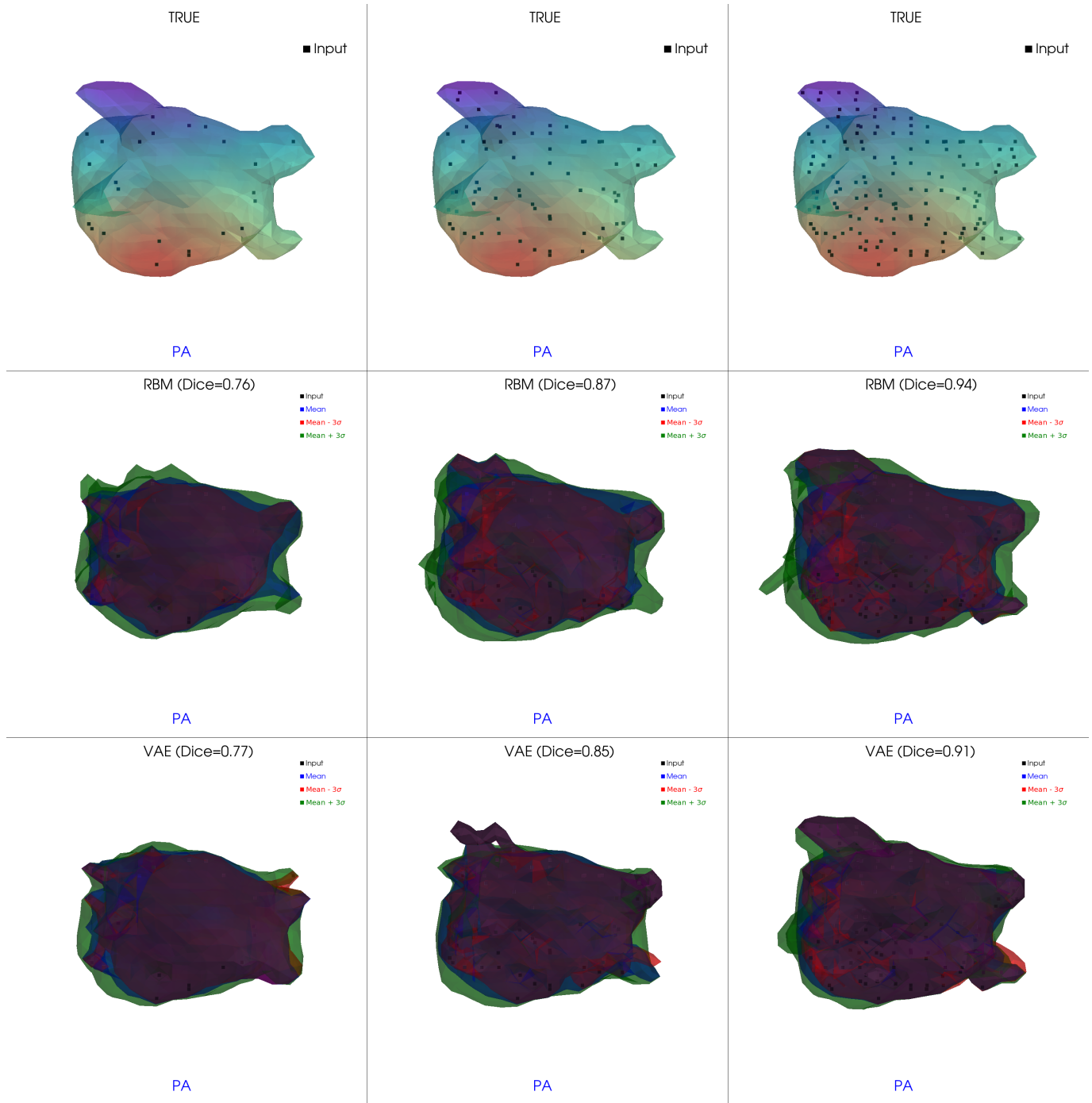
10

Figure 12: RBM, VAE uncertainty comparison with 20, 50 and 100 points in PA view.

chamber compared to the VAE system even though they don't differ much in dice scores. As expected and seen from the dice scores in table 1, the reconstructions are better with more points acquired.

The Figure 12 shows the uncertainty of the RBM, VAE model outputs in the Posterior-Anterior (PA) view. 100 samples were generated to reconstruct the mean, mean +/- standard deviation surfaces. It can be observed that both RBM, VAE models exhibit more uncertainty in the areas where fewer points were acquired. There is also high uncertainty in areas where the pulmonary veins are located in the model. This is due to the high variability of pulmonary veins across patients in the training set.

Table 2 shows task-based assessment of the surface reconstructions of the five patient datasets by three expert observers. Observers individually scored the visual quality and conformity with the true anatomy of the generated meshes on a scale of 1 to

PREPRINT – GENERATION OF PATIENT SPECIFIC CARDIAC CHAMBER MODELS USING GENERATIVE NEURAL NETWORKS UNDER A BAYESIAN FRAMEWORK FOR ELECTROANATOMICAL MAPPING

11

Table 2: Task-based assessment of five patient datasets by three expert observers.

Patient data	RBM						VAE					
	Visual quality			Match w true			Visual quality			Match w true		
	25	100	250	25	100	250	25	100	250	25	100	250
1	3.0	3.3	4.0	2.3	3.6	4.6	2.3	3.3	3.3	2.0	3.6	5.0
2	3.6	3.6	4.0	3.0	3.3	4.6	2.6	3.3	4.3	3.3	3.0	4.0
3	3.0	3.3	4.6	2.0	3.3	4.0	2.3	3.0	3.3	2.3	3.6	4.3
4	3.0	3.0	5.0	2.6	3.6	3.6	2.0	4.0	3.6	3.0	3.3	3.6
5	2.3	3.6	4.3	2.0	4.0	4.3	1.6	3.0	4.0	2.3	3.0	4.0
Average	3.0	3.4	4.4	2.4	3.6	4.2	2.2	3.3	3.7	2.6	3.3	4.2

5, where 1 is the worst possible score and 5 is the best possible score. The table shows the average score of the three observers for the two systems. The table shows that both systems are capable of producing useful reconstructions with the RBM based system performing slightly better than the VAE based system. The table also shows that the reconstructions improve with the number of points acquired.

6 CONCLUSION

Generating anatomical models of the left atrium from a sparse point cloud of acquired locations is a challenging task, but one that is essential for Electrophysiology studies. A probabilistic machine learning approach to the problem provides a way to bring uncertainty quantification and interpretability to a neural network solution to the problem. A Bayesian approach provides greater insight into the model which incorporates prior knowledge from CT/MRI data. Such a model can be useful for navigation during electroanatomical mapping.

The results of the conducted experiments show that the proposed system is capable of generating anatomically accurate models of the left atrium from a sparse point cloud of few acquired locations. This can enable creating quick maps during electrophysiology studies and reduce the time taken for mapping during a study, thereby reducing the patient's exposure to radiation from the fluoroscope.

7 DECLARATIONS

Conflict of interest

The authors declare that they have no conflict of interest.

REFERENCES

, 2021. Voxel-based representation of 3d point clouds: Methods, applications, and its potential use in the construction industry. Automation in Construction 126, 103675. doi:https://doi.org/10.1016/j.autcon.2021.103675.

Colilla, S., Crow, A., Petkun, W., Singer, D.E., Simon, T., Liu, X., 2013. Estimates of current and future incidence and prevalence of atrial fibrillation in the u.s. adult population. The American Journal of Cardiology 112, 1142–1147. doi:10.1016/j.amjcard.2013.05.063. pMID:23831166.

Edelsbrunner, H., Mücke, E.P., 1994. Three-dimensional alpha shapes. ACM Trans. Graph. 13, 43–72. URL: https://doi.org/10.1145/174462.156635, doi:10.1145/174462.156635.

Freund, Y., Haussler, D., 1991. Unsupervised learning of distributions on binary vectors using two layer networks, in: Moody, J., Hanson, S., Lippmann, R. (Eds.), Advances in Neural Information Processing Systems, Morgan-Kaufmann. URL: https://proceedings.neurips.cc/paper/1991/file/33e8075e9970de0cfea955afd4644bb2-Paper.pdf.

Gepstein, L., Hayam, G., Ben-Haim, S.A., 1997. A novel method for nonfluoroscopic catheter-based electroanatomical mapping of the heart. Circulation 95, 1611–1622. URL: https://www.ahajournals.org/doi/abs/10.1161/01.CIR.95.6.1611, doi:10.1161/01.CIR.95.6.1611, arXiv:https://www.ahajournals.org/doi/pdf/10.1161/01.CIR

Hinton, G.E., 2002. Training products of experts by minimizing contrastive divergence. Neural Comput. 14, 1771–1800. URL: https://doi.org/10.1162/089976602760128018, doi:10.1162/089976602760128018.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. science 313, 504–507.

Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K., 1999. An introduction to variational methods for graphical models. Machine learning 37, 183–233.

Kingma, D.P., Welling, M., 2014. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 .

Krum, D., Hare, J., Gilbert, C., Choudhuri, I., Mori, N., Sra, J., 2013. Left atrial anatomy in patients undergoing ablation for atrial fibrillation. Journal of Atrial Fibrillation 5.

Lorensen, W.E., Cline, H.E., 1987. Marching cubes: A high resolution 3d surface construction algorithm. ACM siggraph computer graphics 21, 163–169. URL: https://doi.org/10.1145/37402.37422, doi:10.1145/37402.37422.

Mukherjee, J.M., Mukherjee, A., Mathew, S., Krum, D., Sra, J., 2014. Generation of patient-specific 3d cardiac chamber models for real-time guidance in cardiac ablation procedures, in: Linguraru, M.G., Oyarzun Laura, C., Shekhar, R., Wesarg, S., González Ballester, M.Á., Drechsler, K., Sato, Y., Erdt, M. (Eds.), Clinical Image-Based Procedures. Translational Research in Medical Imaging, Springer International Publishing, Cham. pp. 50–58.

Peterson, G.E., Anderson, C.R., 1987. A mean field theory learning algorithm for neural networks. Complex Systems 1, 995–1019.

Rezende, D.J., Mohamed, S., Wierstra, D., 2014. Stochastic backpropagation and approximate inference in deep generative models, in: Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 1278–1286.

Simpson, A.L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., van Ginneken, B., Kopp-Schneider, A., Landman, B.A., Litjens, G., Menze, B.H., Ronneberger, O., Summers, R.M., Bilic, P., Christ, P.F., Do, R.K.G., Gollub, M., Golia-Pernicka, J., Heckers, S., Jarnagin, W.R., McHugo, M., Napel, S., Vorontsov, E., Maier-Hein, L., Cardoso, M.J., 2019. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. CoRR abs/1902.09063. URL: http://arxiv.org/abs/1902.09063, arXiv:1902.09063.

Smolensky, P., 1986. Information processing in dynamical
    systems: Foundations of harmony theory. Parallel Distributed
    Process 1.

Sra, J., Krum, D., Choudhuri, I., Belanger, B., Palma, M., Brod-
    nick, D., Rowe, D.B., 2016. Identifying the third dimension
    in 2d fluoroscopy to create 3d cardiac maps. JCI insight 1.

Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.,
    2016. Learning a probabilistic latent space of object shapes
    via 3d generative-adversarial modeling, in: Proceedings of the
    30th International Conference on Neural Information Process-
    ing Systems, Curran Associates Inc., Red Hook, NY, USA. p.
    82–90.