

A cartesian grid representation of left atrial appendages for deep learning estimation of thrombogenic risk predictors

César Acebes Pinilla



Universitat
Pompeu Fabra
Barcelona

Bachelor thesis on Biomedical Engineering

Universitat Pompeu Fabra

A cartesian grid representation of left atrial
appendages for deep learning estimation of
thrombogenic risk predictors

César Acebes Pinilla

Supervisor: Dr. Óscar Cámara Rey

Co-supervisor: Xabier Morales Ferez

Department of Information and Communication Technologies

July 2020



Acknowledgments

I would like to express my sincere gratitude to:

- My supervisor, Óscar Cámara, for proposing me this project and believing in me.
- My co-supervisor, Xabier Morales, for putting up with me and always being willing to help me.
- My former professor Jordi Mill, for teaching me about the Left Atrial Appendage during last year subjects MOS and Computational Cardiology.
- My fellow degree student, Mariabel, for helping me at the first steps of the Thesis project
- My family and friends, for supporting me in very moment.

Abstract

Atrial fibrillation is the most common sustained arrhythmia in the world. This condition is responsible for about 20% of the cardioembolic ischemic strokes, of which, most are caused by thrombus formed in the Left Atrial Appendage (LAA). Computational fluid simulations have been used to assess hemodynamic implications of AF on a patient-specific basis. Deep Learning (DL) has shown potential in accelerating these simulations, although many of the most successful algorithms such as Convolutional Neural Networks (CNN) rely on the Euclidean structure of the data. Therefore, the aim of this study consisted on generating a fast surrogate of fluid simulations, that predicted the thrombus formation risk. For this purpose, a new flattened representation of the LAA was achieved by sampling the LAA in two directions: from the junction to the left atrium (i.e. ostium) to the tip (i.e. apex), using the normalized gradient of the heat flow and radially. It showed a great potential for clinical use and representation of LAA structures as a two-dimensional flat geometry as well. Using the node discretization provided by the flattening algorithm, two Deep-Neural Networks (DNN) and one CNN configurations were tested. The mean absolute errors on the thrombogenic risk index given by the DL configurations were similar (0.74 and 0.73) while it was lower for the CNN (0.63). Similarly, the CNN-based approach detected better the highly thrombogenic risk areas compared to the DNN-based ones (87.9% vs 81.0% and 81.7%, respectively).

Keywords

Keywords: deep learning; modelling; mesh processing; left atrial appendage; conformal flattening; computational fluid simulations; thrombus risk predictors.

INDEX

	Page.
1. Introduction	1
1.1. Atrial fibrillation	1
1.1.1. Pathology	1
1.1.2. Characterizing the thrombus development risk	2
1.2. Fluid simulations to predict thrombogenic risk	3
1.3. Deep Learning	3
1.4. Objectives of the present study	5
2. Methods	6
2.1. Geometric dataset and simulations	6
2.2. Flattening algorithm	8
2.2.1. Radial mapping	9
2.2.2. Angular mapping	10
2.3. Architecture of the Deep Learning models	12
2.3.1. Liang model	12
2.3.2. Morales model	15
2.3.3. Convolutional Neural Network-based model	16
2.4. Training and evaluation of the Deep Learning models	17
2.4.1. Training of the Deep Learning models	17
2.4.2. Evaluation of the Deep Learning models	18
3. Results	20
3.1. Flattening algorithm	20
3.2. Sensitivity analysis	21
3.3. Performance of the Deep Learning models	22
4. Discussion	26
5. Conclusions and future directions	29
List of Figures	30
List of Tables	34
Bibliography	35
Appendix	39

1. INTRODUCTION

1.1 Atrial Fibrillation

1.1.1 Pathology

Atrial fibrillation (AF) is the most common sustained arrhythmia encountered in clinical practice, mostly prevalent among elderly men [1, 2]. In this pathology, the atria, which are the upper chambers of the heart, quiver or beat irregularly instead of beating effectively to move blood into the ventricles, leading to clot formation, stroke, heart failure or other heart-related complications [3].

Nowadays, the risk of clot formation is treated by oral anticoagulation, such as Warfarin. However, if the patients suffer from bleeding or the medication is too risky for them, a suitable alternative is the Left Atrial Appendage Occlusion. These devices completely seal the Left Atrial Appendage (LAA), preventing blood from entering and reducing the risk of thrombi formation, as it can be seen in Figure 1.

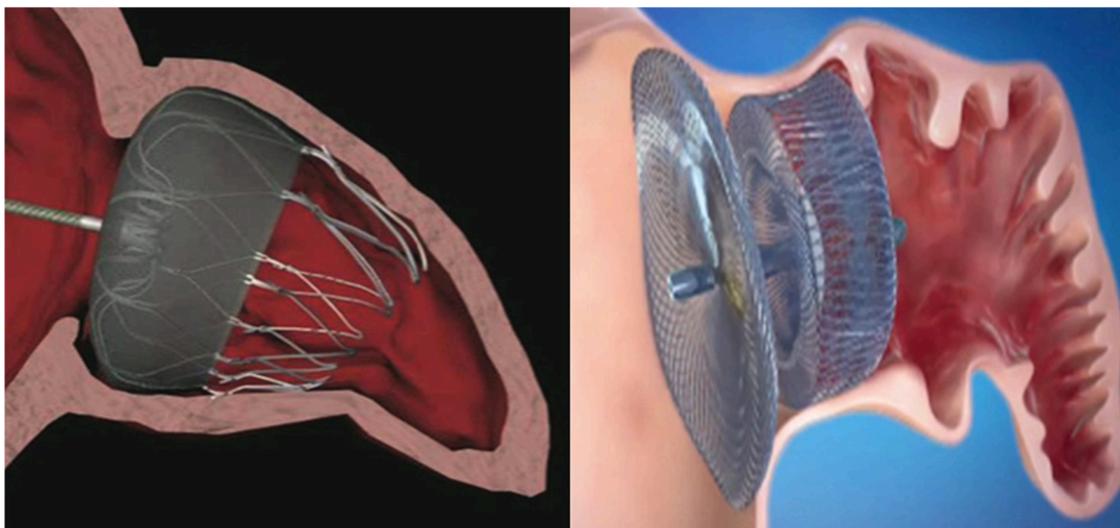


Figure 1. Watchman™ and Amplatzer™ Cardiac Plug devices after implantation in the LAA. From [4].

As a matter of fact, AF is responsible for 15 to 20% [5] of all cardioembolic ischemic strokes, more of the 90% of which are caused by thrombus formed in the LAA, a finger-like cavity located in the left atria (LA). It is a bulged structure derived from the anterior wall of the LA with a hooked tubular shape. Its junction with the LAA is well defined by the ostium, a narrowing of the appendage's orifice. This structure is highly heterogeneous and presents a high inter-patient variability in its size and shape, as presented in Figure 2.

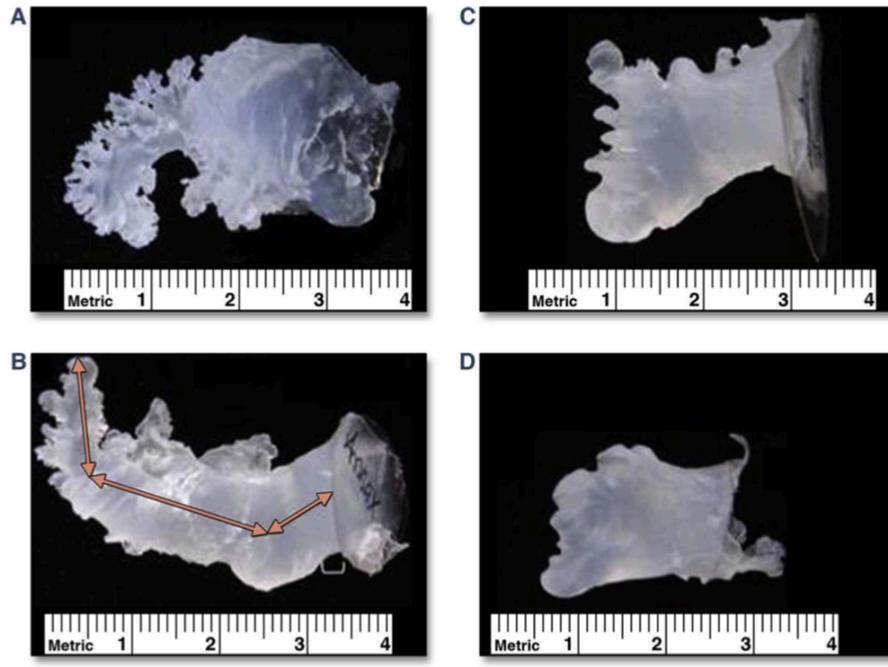


Figure 2. Endocasts obtained from explanted hearts showing the different LAA intraluminal morphologies. From [6].

The influence of the LAA hemodynamics on thrombus development is a widely investigated topic [6, 7, 8]. Conditions such as reduced blood flow velocity in the LAA have been linked to an increased risk of thrombus formation [9], while others, such as presenting of a specific type of LAA morphology have also been hypothesized to play a key role in clotting. Nevertheless, according to recent studies, inconsistent LAA morphology classification coupled with low interobserver and intraobserver agreement suggests caution when directly linking LAA morphology to the risk of stroke [10].

Computational Fluid Dynamics (CFD) has been widely employed on the study of LA thrombogenesis over the past few years [11, 12, 13]. In fact, it has proven to be an invaluable tool in establishing a mechanistic relation between patient-specific LA morphologies and their characteristic hemodynamics.

1.1.2 Characterizing the thrombus development risk

The process of thrombi development is linked to a combination of several hemodynamic conditions. Velocities under 40 cm/s have been associated with higher stroke risk and those under 20 cm/s with the identification of blood clots in the LAA [6]. Therefore, LA hemodynamic analysis becomes a fundamental tool in determining the risk of thrombi formation. So far, these data have been evaluated through noisy image data of transesophageal echocardiography. Unfortunately, it oversimplifies the fourth-dimensional nature of blood flow patterns, as it can provide two-dimensional (2D)

velocities at most. One way to characterizing the complex blood hemodynamics of the cardiac chambers is time-resolved three-dimensional phase contrast magnetic resonance imaging, also known as 3D Cine PC-MRI or 4D Flow MRI, more colloquially, since it is able to obtain the full three-dimensional (3D) velocity vector field along time. Nevertheless, as this technique is still under development, it faces several challenges such as low spatial resolution which hampers its ability to compute more advanced flow parameters accurately.

1.2 Fluid simulations to predict thrombogenic risk

CFD is a branch of fluid mechanics, that consists on numerically analyzing fluid flow. It has been used in order to determine the most important LA and LAA morphological parameters in atrial flow dynamics [9], and to assess the complex flux characterization parameters, such as the Wall Shear Stress (WSS) [14] or the Endothelial Cell Activation Potential (ECAP). The ECAP is the principal factor that characterizes the thrombogenic susceptibility of a vessel, according to di Achille *et al.* [8], since it detects regions exposed to both low and oscillatory shear flow. High values of ECAP in an area are correlated well with locations of thrombus, so it allows identifying areas with a high risk of thrombosis.

In addition, CFD has allowed the development of LA models to learn hemodynamic implications of AF on a patient-specific basis [13]. However, CFD presents a drawback, which is the vast computational resources and long computation times that they require to complete. It poses a significant limitation in the use of computational models in clinical applications.

1.3 Deep Learning (DL)

The substantial growth of computational power and data storage capabilities in recent years has resulted in a change of paradigm in the DL, allowing its widespread use in every single research field in medicine [15]. Nevertheless, it has not been until recently that DL algorithms have been employed in the high dimensional complex dynamic systems, such as fluid dynamics [16, 17]. Consequently, several publications have tried to study biological fluid flows, aiming to alleviate the time burden imposed by regular CFD simulations.

In this regard, Liang *et al.* [18], in 2018, suggested a Deep Neural Network (DNN) based method to directly estimate the WSS of the aorta, without performing the finite element analysis. Based on its work, Morales *et al.* [19], in 2019, designed a DNN with the objective of generating a fast surrogate of the LAA, to quickly assess the risk of thrombus formation in the LAA, without needing to perform the fluid simulations. However, his work presented some limitations, especially in the pre-processing stage of the meshes used for training the network. In particular, all LAAs needed to be composed of the same number of nodes sharing the same inter-nodal connectivity, achieved by registering each geometry to a template. In some points, it generated coarse triangle elements, that needed to be manually smoothed through visual inspection for each LAA. Moreover, there were registration errors due to large differences not recovered by the surface registration procedure and finally, the LAA had to be assembled to the LA and aligned in a very specific manner, ensuring the correct orientation. In addition to this, as the registration was based on a non-Cartesian representation, it only allowed fully connected networks to work; impeding the use of novel neural network architectures such as Convolutional Neural Networks (CNNs), which have proven to be very efficient in image processing tasks. CNNs are artificial neural networks that take in an input image and reduce it into a form which is easier to process, without losing important features [20]. There are many examples of the application of CNNs in the clinical world, some of which can be seen in the review study by Anwar *et al.* [21].

Thereby, a practical way of obtaining a Euclidean representation of the data is required, so that it can be fed to the mentioned algorithm. One such approach would consist on performing a flattening of the LAA geometry, which is a parametrization technique that consists on mapping 3D data into the 2D plane [22, 23]. After it, 3D structures can be seen entirely in a single image, thus making them easier to understand. In medicine, flattening is widely used, since it facilitates visualization of complex structures and comparison among patients, in order to determine physiological aspects. Some of the applications in the clinical practice can be seen in the work of Kreiser *et al.* [22].

A recurrent example of techniques used for visualizing flat structures consists on mapping the 3D structure into cartesian grid-based circular plots. It is a technique previously studied in cardiology, in the work of Köhler *et al.*, that mapped 4D PC-MRI into circular plots [24] or in the Cardiac Bull's Eye Plot, proposed by Cerqueira *et al.*, in 2002 [25]. This representation has been widely adopted in the field of Cardiology to characterize the

17 myocardial segments of the left ventricle. Figure 3 shows two examples of the aforementioned representations.

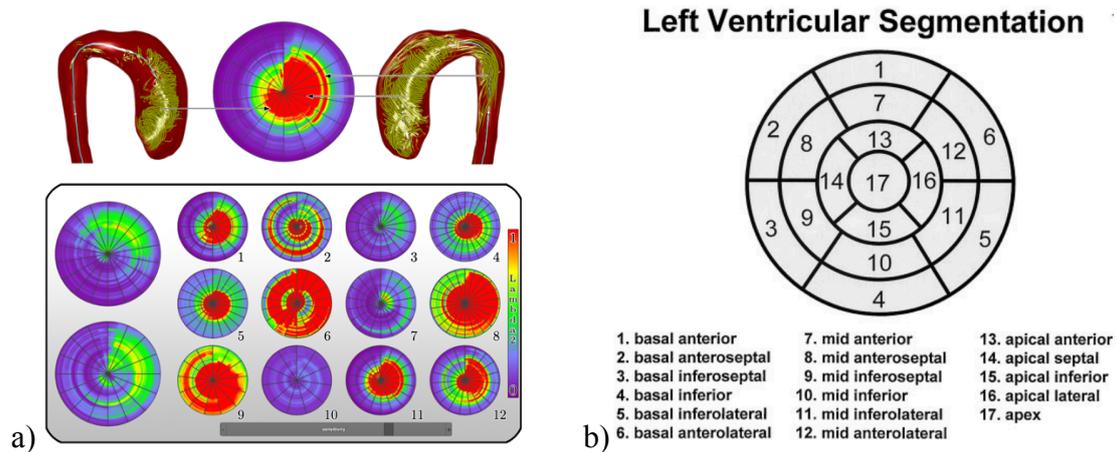


Figure 3. Flat representation of the aorta (a), from [24] and Cardiac Bull's Eye Plot (b), from [25].

This way, the representation of the LAA as a cartesian grid would allow for the employment of CNNs.

1.4 Objectives of the present study

Morales *et al.* study [19] has shown great potential in predicting ECAP distribution in the LAA. However, as stated before, it has presented several drawbacks when was implemented. One of the fundamental limitations is that it did not allow CNN-based models to be applied, since the LAA population could not be represented as Euclidean datasets.

For this reason, the goal of the present study was to generate a fast surrogate of fluid simulations, based on CNN, to predict the thrombus formation risk. To achieve this goal, it was necessary to develop an established standard representation of the LAA, achieved by mapping its surface to a 2D image through a novel algorithm. Afterwards, the flattened representation of the LAA was used to train state-of-the-art DL models, including based on CNN, to learn the relationship between the patient-specific LAA geometry and the in-silico risk predictors of thrombus formation risk.

2. METHODS

The methodology of this study is divided into two different sections. Firstly, the creation of the dataset that contains all the training data, i.e. the 2D representations of the LAA geometries, using previous CFD simulations. Additionally, the description of the DL algorithms used and the error metrics for assessing their accuracy are covered. Figure 4 shows the workflow of the project.

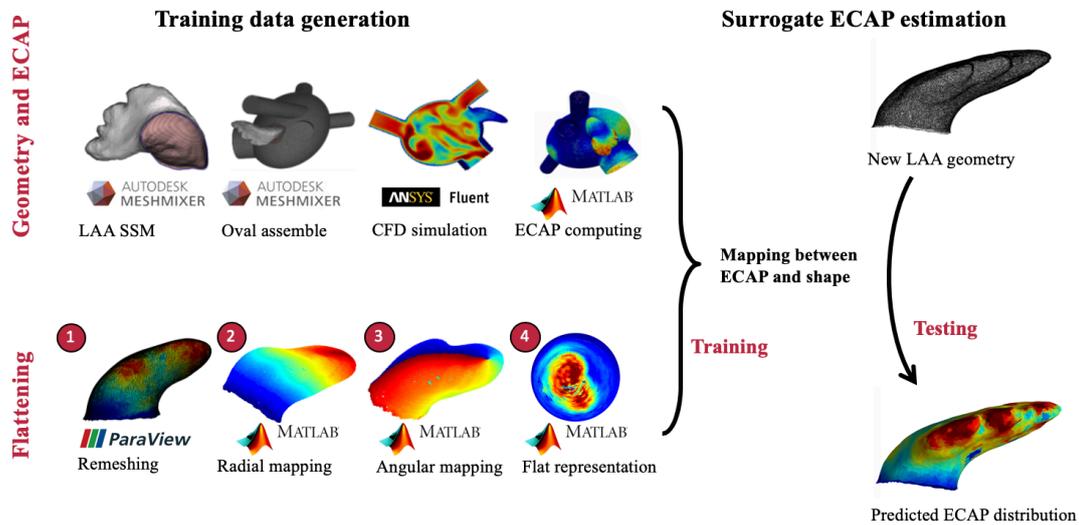


Figure 4. General pipeline of the study. The four steps related to geometry and ECAP obtaining were performed by Morales et al. [19]. Following the enumeration, the LAA are remeshed (1) and introduced into the flattening algorithm, where a radial (2) and an angular (3) mappings are applied over them, to obtain the flat representation (4). Afterwards, the DL networks are trained to estimate the ECAP in new LAA geometries. LAA: Left Atrial Appendage, SSM: Statistical Shape Model, CFD: Computational Fluid Dynamics, ECAP: Endothelial Cell Activation Potential.

2.1. Geometry dataset and simulations

The input shapes used in this work consisted on a virtual dataset of 300 artificial LAA geometries, obtained from a Statistical Shape Model (SSM) provided by Slipsager *et al.* [26]. In that study, non-rigid volumetric registration of signed distance fields was applied to 103 patient LAA surfaces obtained from the Department of Radiology, Rigshospitalet, University of Copenhagen. This ensured the same inter-nodal connectivity and number of nodes in all geometries. Subsequently, the SSM was generated based on Principal Component Analysis (henceforth referred to as PCA).

After obtaining the data, the LAAs were attached to an oval using Meshmixer¹, which acts as a simplification of the LA, with based on the work by García-Isla *et al.* [9]. This

¹ <http://www.meshmixer.com/>

complexity reduction was necessary to narrow down the complexity of the study by limiting the geometry variations of each case to the LAA shape. Then, the volumetric mesh was created using Gmsh² and afterwards introduced to ANSYS-Fluent³. A whole cardiac cycle was simulated for each case, divided into systole and diastole, setting up the parameters according to the study by Mill *et al.* [27]. After the simulations, the parameters necessary to compute the ECAP were exported. Finally, this calculation was made in MATLAB 2019b⁴ for each element of the mesh [9]. In this study, the ECAP was used to detect thrombogenic susceptible areas, since it unifies several parameters related to the pro-thrombogenic endothelial susceptibility.

The final virtual dataset finally consisted on 206 artificial LAA geometries with the ECAP values in each node. Some of the simulations were discarded, since they showed unrealistic results.

To more accurately represent the LAA mesh, the geometries were pre-processed. They were exported, firstly, to Meshmixer, in stl format. There, they were manually remeshed, increasing the number of nodes in each mesh, from 2466 to ~18,500.

Afterwards, in order to ensure that each node had its correspondent value of ECAP, the remeshed stl file was exported to Paraview⁵, where an interpolation filter that used the Voronoi kernel was applied. This filtering is also known as nearest-neighbor interpolation. It allowed each node to have an ECAP value, calculated after interpolating the original ECAP values. Figure 5 compares the geometry of a LAA before and after remeshing it, there can be seen the increase of triangle elements without losing the spatial information of ECAP.

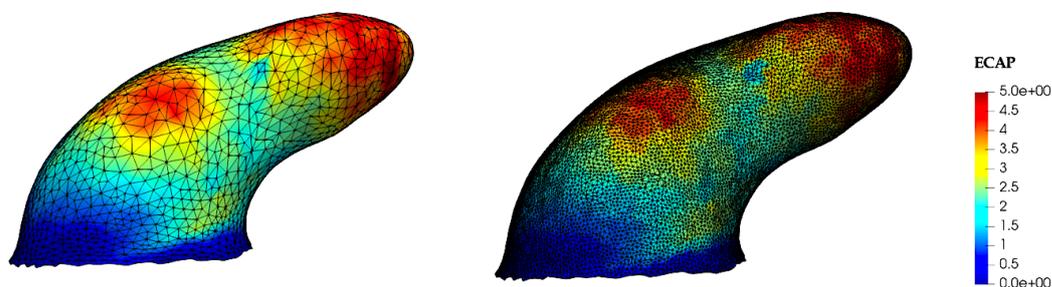


Figure 5. Comparison of geometries before and after performing the remeshing. In the left, original LAA; in the right, manually remeshed LAA. ECAP: Endothelial Cell Activation Potential.

² <http://gmsh.info>

³ <https://www.ansys.com/products/fluids/ansys-fluent>

⁴ https://www.mathworks.com/products/matlab.html?s_tid=hp_products_matlab

⁵ <https://www.paraview.org/>

2.2. Flattening algorithm

The mapping of the LAA from the 3D geometry to the 2D representation was applied by calculating polar-based coordinates, using a method similar to that proposed by Meuschke *et al.* [28] for mapping the aneurysm surface to a hemisphere.

The flattening algorithm consisted of two successive steps, which were the radial mapping and the angular mapping. They were used to select the nodes of interest to discretize the contour from the 3D geometry. Figure 6 shows a schematic of the flattening algorithm.

The algorithm input was made up by the geometry of the LAA, the desired number of radial and angular isolines, the coordinates of the circumflex artery and by the timestep. The geometry of the LAA consisted on the spatial coordinates of the nodes and the correspondences among them, which are the two necessary requirements to create a mesh. The number of radial and angular isolines depended on the wanted resolution of the planar representation. The coordinates of the circumflex artery consisted only in the three spatial coordinates of a point, necessary to orientate the 2D representation. The selection process of the optimal timestep is explained later.

The selected representation for the output of the algorithm, for its visual ease and topology, was a bull's eye plot. It is a representation based on a circular grid, in which each radial isoline is equivalent to a circumference; considering this, each sample of that isoline has the same radius. On the other hand, each radial isoline is equivalent to a radius of the circle, as seen in Figure 6.

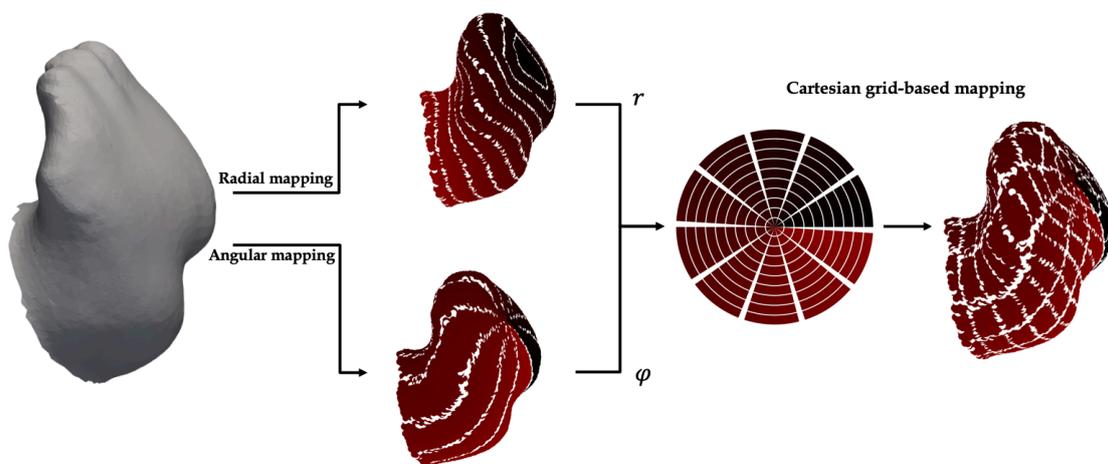


Figure 6. A schematic for illustrating the calculation of polar-based coordinates. From left to right: 3D mesh, which is the input of the algorithm; radial (top) and angular (bottom) isolines obtained from radial and angular mapping, respectively, flat representation of the cartesian grid and corresponding nodes in the 3D geometry. Adapted from Meuschke *et al.* [28].

2.2.1. Radial mapping

In order to obtain the radial mapping, the algorithm started detecting the nodes that form the border of the mesh, which correspond to the ostium of the LAA. After that, the geodesic distance between these nodes and the rest of them was computed.

There is a large number of approaches to calculate it, but the one chosen due to its speed was the Heat Method, proposed by Crane *et al.* in 2013 [29] and applied in MATLAB 2019b⁶. This method consists of three basic steps, which are, firstly, the integration of the heat flow $\dot{u} = \Delta u$ for some fixed time; then, the evaluation of the vector field X , which is $X = -\nabla u / |\nabla u|$ and finally, the solving of Poisson equation $\Delta \varphi = \nabla \cdot X$. Here, X is the normalized gradient field of ∇u and φ is the approximation of geodesic distance, as timestep goes to zero. The timestep t was chosen according to the numerical experiments carried out by Crane *et al.*, stating that $t = mh^2$, where h is defined as the mean spacing between adjacent nodes and $m = 1$, for near-optimal accuracy. In the case of this study, h was the average edge length of the mesh, which was 0.001, so t was 10^{-6} . The final radial mapping was then normalized, yielding distances between 0 and 1.

After applying the Heat Method, the nodes of the desired number of isolines were selected. Isolines are groups of nodes that are at the same distance from the LAA ostium and they were distributed regularly along the LAA from the ostium to the most distant node, which is the so-called apex. Figure 7 displays the effect of choosing a different number of isolines and how they are evenly distributed lengthwise.

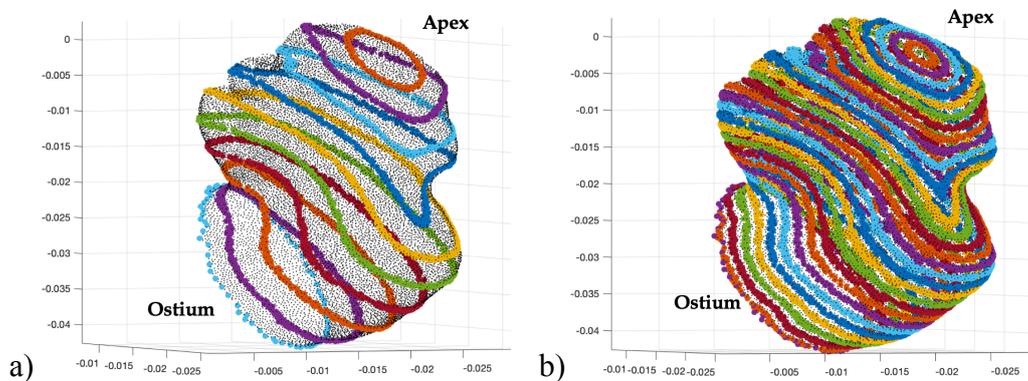


Figure 7. Isolines of the LAA. (a) 10 isolines, (b) 30 isolines. In black, the nodes of the LAA. Colored, the nodes corresponding to the isolines.

⁶ Using code from http://www.numerical-tours.com/matlab/meshproc_7_geodesic_poisson/
Retrieved May 17, 2020.

Afterwards, a plane was calculated for each isoline, by performing a linear regression taking as input the nodes of each. The code for performing the planar fit was retrieved from a virtual MATLAB forum⁷. After that, each node was projected into the plane corresponding to its isoline, so that each isoline lied in a plane. Figure 8 shows the nodes that compose 5 isolines and their projection into their respective plane.

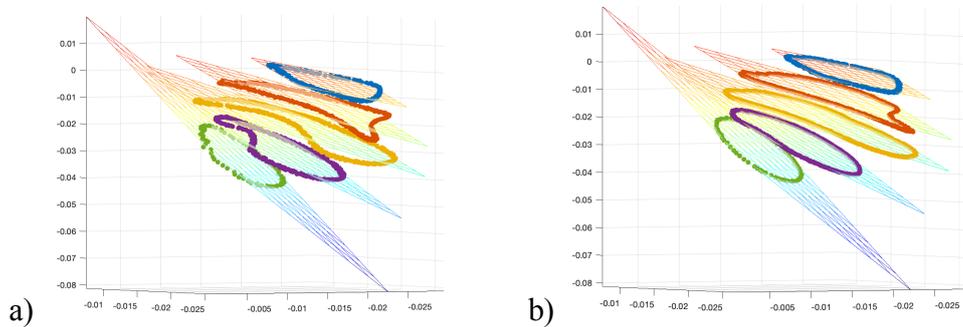


Figure 8. Nodes of the isolines before (a) and after (b) being projected into their respective 3D planes.

2.2.2. Angular mapping

The objective of the angular mapping was to sort the nodes of each isoline by the angular distance they have with respect to a reference node. In order to determine what is the mentioned node, the circumflex artery was considered, since it is a structure that is close to the LAA. To determine the reference node of each isoline, the furthest node from the circumflex artery was selected. In case of selecting the closest node as reference, some errors would happen in some isolines, making the angular mapping unprecise. After finding the reference nodes, the center of each isoline was computed. Afterwards, the angular distance of each node, relating to the reference one was calculated by means of the following mathematical operations:

Stating $v1 = ref_n - center$ and $v2 = node' - center$, where ref_n is the projection of the reference node, $node'$ is the projection of a determined node N in the plane and $center$ is the center of that plane, the cross product $x = v1 \times v2$ was computed, calling its normal nx . With these data, the angular distance A between N and the reference node was computed according to Equation 1:

⁷ https://es.mathworks.com/matlabcentral/answers/448708-plane-fitting-a-3d-scatter-plot#answer_364205. Retrieved May 17, 2020.

$$A = \tan^{-1}(nx, v1 \cdot v2) \quad [1]$$

In it, \cdot and \tan^{-1} , represent the dot product and the four-quadrant inverse tangent, respectively. Since the aforementioned process yields angular distance values between $-\pi$ and π , they were operated to be between 0 and 2π .

Then, the nodes in each isoline were sorted by its angle, to select the nodes of the angular isolines. For it, the angle with the closest value to the angle of interest was selected, depending on the desired number of angular isolines. Figure 9 shows the graphic representation of both angular and radial mappings.

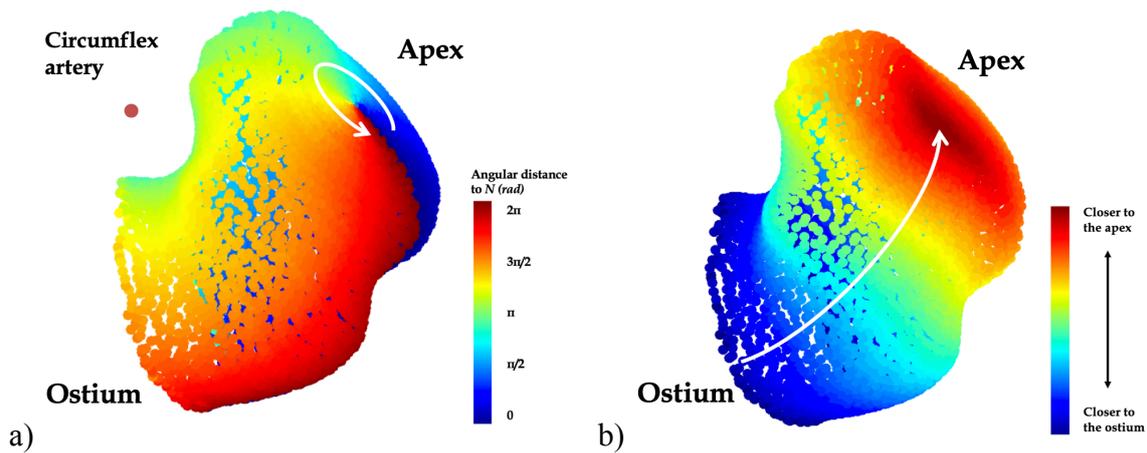


Figure 9. Graphic representation of the angular (a) and radial (b) mapping. In (a), nodes are colored according to the angular distance to the reference node. The white arrow crosses the different nodes of a radial isoline, with increasing angular distance values. In (b), nodes are colored according to the distance to the ostium. The white arrow crosses the different nodes of an angular isoline, with increasing distance to the ostium. N is the reference node of each radial isoline, diametrically opposite the circumflex artery.

Thereby, once all the nodes of the contour discretization were chosen, their coordinates were stored in a matrix, where each row was an angular isoline and each column was a radial isoline. The ECAP value of each node was likewise stored in a different array, so it could be graphed later, as shown in Figure 10. The code used for achieving the bull's eye plot representation was retrieved from a virtual MATLAB forum⁸ and used with minor changes.

⁸ Daniel Ennis (2020). Bullseye Polar Data Plot (<https://www.mathworks.com/matlabcentral/fileexchange/16458-bullseye-polar-data-plot>), MATLAB Central File Exchange. Retrieved May 17, 2020.

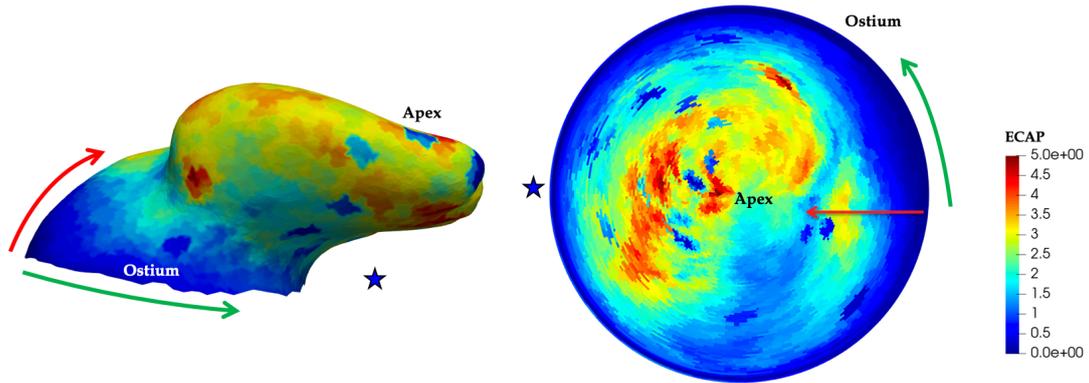


Figure 10. 3D structure and flat representation of the LAA achieved in this study. Direction of radial and angular mapping is displayed, in red and green, respectively. The apex and a point of the ostium are emphasized. The blue star represents the relative position of the circumflex artery with respect to the LAA. ECAP: Endothelial Cell Activation Potential.

2.3 Architecture of the Deep Learning models

This work develops a surrogate of CFD simulations, accelerating the process of obtaining the ECAP by means of three DL models, which are tested. The objective was to directly estimate the ECAP maps of the LAA given its shape. For this task, the dataset composed of 206 LAA ECAP maps obtained from the discretization and their respective 3D coordinates of that points was used.

The different DL approaches that have been studied are detailed in the following sections. Firstly, the two DNN-based ones (henceforward referred to as Liang model and Morales model, respectively) and then, the CNN-based model.

2.3.1 Liang model

The first model used was a DNN, adapted from Liang *et al.* [18], as Morales *et al.* was not able to implement it due to its representation constraints. Figure 11 presents a diagram of this model, which is basically composed of three main steps: the shape encoding, the non-linear mapping and the ECAP decoding.

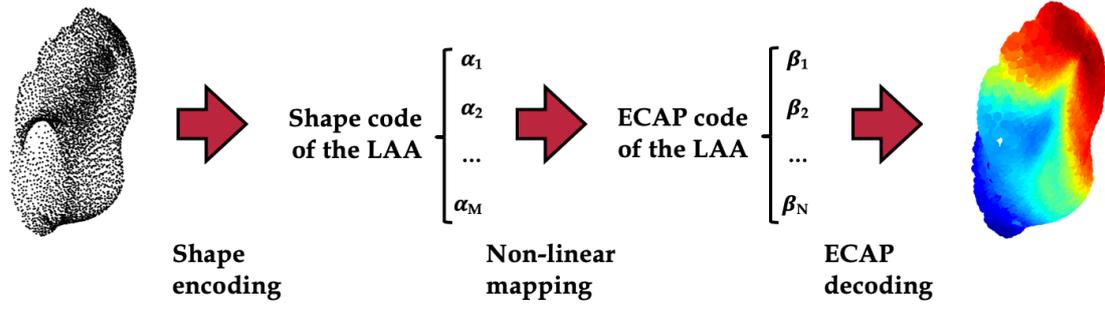


Figure 11. Overview of the DNN workflow, adapted from Liang et al. [18]. The input to the network is the shape of the LAA, represented by the coordinates of the nodes and the output is the ECAP distribution in each node. α_{1-M} and β_{1-N} are the components of the shape code and of the ECAP code of the LAA, respectively.

Shape encoding was the first step, meaning that spatial coordinates of the LAA shape, represented by a quadrilateral mesh, have been encoded into a set of scalars. This process was carried out by means of PCA, which allowed reducing the dimensionality of the training set. It consists on finding the set of modes of variation or Principal Components (PCs) that best describes the observed variation so that any shape can be approximated by a linear combination of these modes [30]. By using PCA, a determined LAA geometry X can be represented as follows:

$$X = \bar{X} + \sum_{m=1}^M \alpha_m \sqrt{\lambda_m} W_m \quad [2]$$

In this equation, \bar{X} is the mean shape of the LAAs: $\bar{X} = \sum_{i=1}^s x_i$; λ_m and W_m are, respectively, the eigenvectors and eigenvalues of the covariance matrix and α_m is the shape code of the LAA.

Parting from Equation 2, the shape code could be represented by using the Equation 3, that has been constructed using the property that establishes that eigenvectors are orthogonal to each other.

$$\alpha_m = \frac{W_m^T (X - \bar{X})}{\sqrt{\lambda_m}} \quad [3]$$

Here, m are the PCs used for encoding the LAA. It is important to emphasize that LAA is a complex structure, so the number of PCs selected must be relatively high, in order to retain a significant overall variability (90%-98%). Figure 12 displays the PCA dimensionality reduction network.

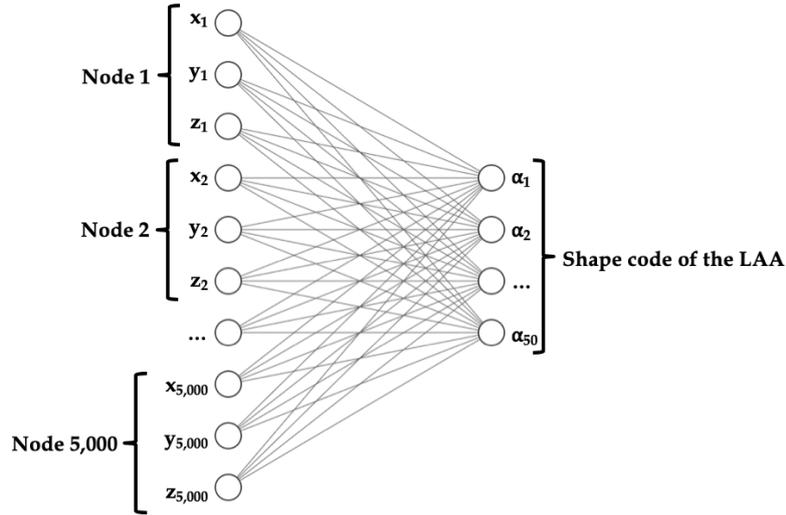


Figure 12. Network used for the shape encoding by means of PCA. x_i , y_i and z_i are the coordinates of the node $i \in (1, 5000)$ and α_{1-20} are the components of the shape code of the LAA.

After encoding the coordinates, a fully-connected neural network was used to map the shape code to the ECAP code. The network that achieved the non-linear mapping is presented in Figure 13. It was comprised of 3 hidden layers, each of them has 64 softplus activation units [31]. Other widely used activation units, such as ReLU, ELU, Parametric ReLU or Leaky ReLU, were also evaluated and discarded, due to the less accuracy they showed.

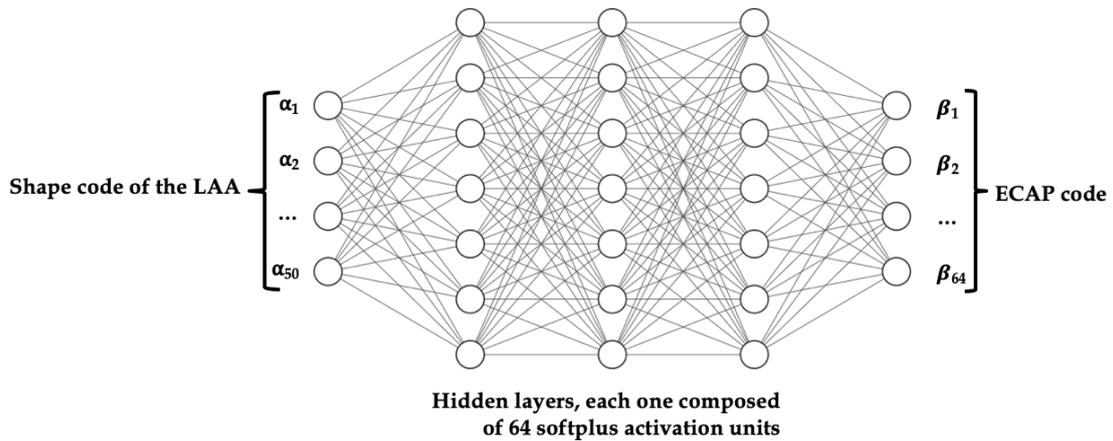


Figure 13. Fully connected neural network used for the non-linear mapping. α_{1-20} and β_{1-64} are the components of the shape code and of the ECAP code, of the LAA, respectively.

Finally, concerning the ECAP decoding, a bidirectional neural network was used. The bidirectionality allowed it to be used in the ECAP decoding as well as in the ECAP

encoding process, by reversing it. However, the encoding was only used in the training step.

The ECAP decoding was based on Low Rank Approximation (LRA) [32]. For decoding the ECAP, the LAA quadrilateral flat wall was divided into 25 regions, each one composed of 200 nodes, therefore, the ECAP code of each region could be calculated from the ECAP code of the whole LAA. Then, the ECAP values of each node were computed from the ECAP code of each region, as it is shown in Figure 14.

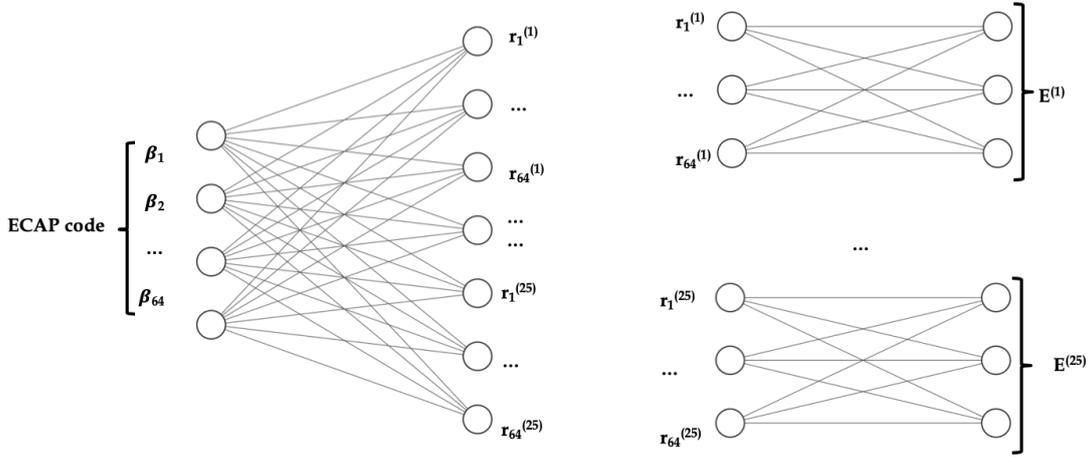


Figure 14. Network used for the ECAP decoding. β_{1-64} are the components of the ECAP code of the LAA, r_{1-64} are the components of the ECAP code of each region, from (1) to (25) and $E^{(1)}$ to $E^{(25)}$ are the values of ECAP of each region.

2.3.2 Morales model

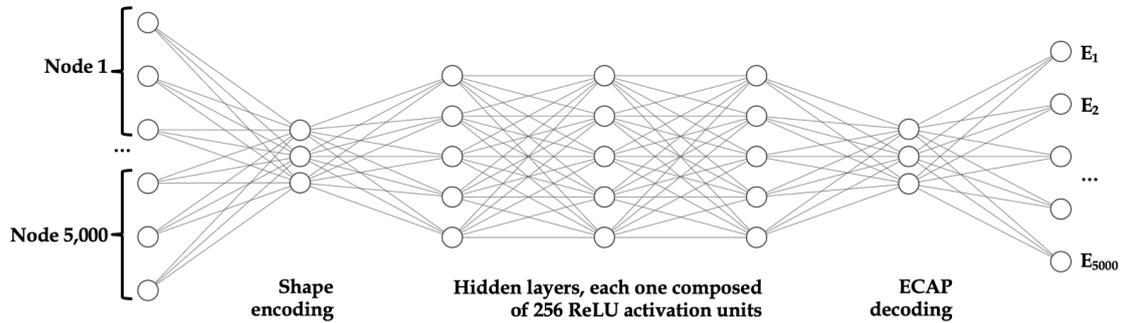


Figure 15. Workflow of the second DL model. The input consisted on the coordinates of the nodes, shape encoding and decoding steps were performed by truncated PCA and the output consisted on the ECAP values of the nodes.

The second DNN-based model was, in this case, adapted from Morales *et al.* [19]. The workflow of this model is presented in Figure 15 and, likewise the previous one, is composed of three steps, which are the shape encoding, the non-linear mapping and the stress decoding.

Firstly, the shape was encoded by truncated PCA, retaining the most significant PCs, followed by a non-linear mapping achieved by a fully-connected feed-forward neural network composed of 3 hidden layers, each one comprised of 256 ReLU activation units. In this model, the most important difference with the previous one is the ECAP encoding and decoding steps, which are done by means of truncated PCA.

2.3.3 Convolutional Neural Network-based model

In addition to the aforementioned DL models based on DNN, another approach was evaluated. It consisted on a CNN-based model inspired by the work of Li *et al.* [33], that predicted the spatial distribution of a reaction-diffusion system using an encoder-decoder based CNN. In that work, the CNN was used as a surrogate of a finite element method. This architecture allows to multiscale feature extraction by combining convolution layers with several pooling operations, where the encoder manages to represent each geometry as a condensed feature map. Later, this compacted feature vector can be decoded in order to produce an output, that could be a segmented image [34] or, as in this case, a map of ECAP. As in the previous DL models, the objective was to directly estimate the ECAP of each node of the LAA given its geometry without needing to compute fluid simulations. Since CNN are networks that work with Euclidean datasets, such as 2D or 3D images, both the input and the output of the network must be a matrix. The flattening algorithm has allowed the ECAP map of the LAA to be depicted in a 2D plane, but, it has also let the coordinates to be represented in that format. In this manner, input dataset was divided into 3 different channels: depicting the x, y and z coordinates of the LAA geometries. A graphical representation of the encoder-decoder CNN is displayed in Figure 16. The input of the network was a three-channel tensor, depicting the three coordinates. This tensor was then processed by the encoder part, where some convolutional operations were used to extract the features and *maxpooling* layers were used to down-sample these features. Afterwards, the decoder allowed to reverse the process in the encoder and to generate the predicted solution.

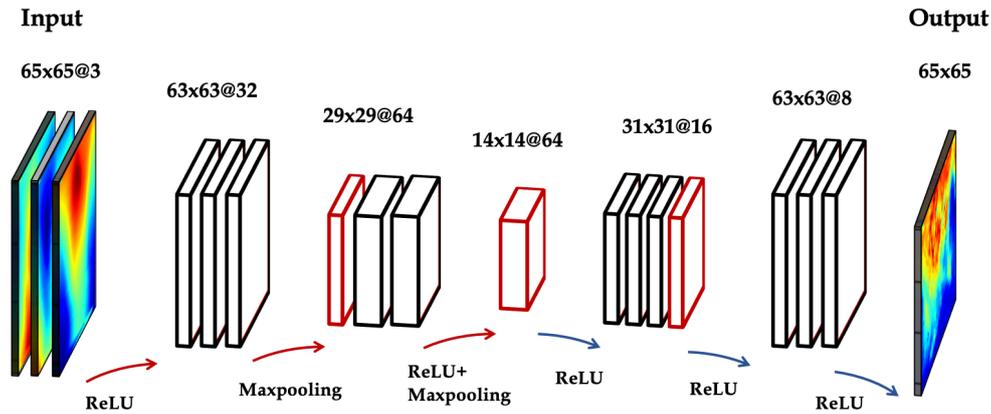


Figure 16. Overall encoder-decoder CNN architecture. Given the features from the encoder part (red arrows), the ECAP prediction was obtained through multiple decoding operations (blue arrows).

2.4 Training and evaluation of the Deep Learning models

2.4.1 Training of the Deep Learning models

The parameters of the three different DL models were learnt from the training data. In the Liang model, the number of PCs, in shape encoding and singular components, obtained through LRA, in ECAP encoding, were firstly selected. The chosen number of PCs was 50, which explained about 98% of the variability. On the other hand, the number of singular components that were chosen was 128 in the first LRA and 64 in the second one, which yielded an approximation error less than 1%. On the subject of the non-linear mapping hyperparameters, the selected optimization algorithm was Adamax [35], the batch size was 64 and the number of epochs was 500.

For Morales model, in shape encoding, the selected number of PCs was 32, similarly to in ECAP encoding and decoding, where also the 32 most significant PCs were retained. This explained a total variability of 97% and 84%, respectively. Adam was the selected optimizer for the fully-connected network that performed the non-linear mapping; the batch size was 20 and the number of epochs was 200.

Finally, for the CNN-based model, the training step was done in 400 epochs, with a batch size of 32 and a learning rate of $5 \cdot 10^{-4}$. This model was trained using the resources provided by Google Colab⁹ environment; in particular, the NVIDIA Tesla P100¹⁰ Graphic Processing Unit or GPU was used, drastically decreasing the training time.

⁹ <https://colab.research.google.com/>

¹⁰ <https://www.nvidia.com/en-us/data-center/tesla-p100/> Accessed June 5, 2020.

In the three networks, the training hyperparameters were tested in order to prevent overfitting. Besides, the selected loss function for all of them was the mean-squared error.

2.4.2 Evaluation of the Deep Learning models

The performance of the three models was then evaluated through Monte Carlo cross-validation. For this, the results obtained after the flattening of the CFD simulations carried out by Morales *et al.* [19] were considered as the ground truth. In Monte Carlo cross-validation, the dataset is randomly divided into two parts: in this case, the 90% of the shapes formed the training set and the remaining 10% formed the test set. In each iteration of the cross-validation, the shapes that comprise the training set are used to fit the DL model, while those in the test set are used to evaluate the accuracy of the prediction.

This accuracy was assessed through MAE, which is the Mean Absolute Error and RMSE, the Root Mean Square Error. MAE is defined as the average of the absolute difference between prediction and ground truth over the test set if all individual differences present the same weight (Equation 4), while RMSE is defined as the square root of the average differences between prediction and ground truth (Equation 5).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad [4]$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad [5]$$

In addition, another technique to assess the predictive capability of the DL models was used, that was based on the fact that higher values of ECAP represent areas with high thrombogenic risk. Following the study of Di Achille *et al.* [8], the lower boundary for this risk was set on a value of 4, that corresponds to the upper 99th percentile, since it is more robust than only considering isolated peak values.

This limit allowed binary classifying the nodes in two categories: Positive, which includes those nodes whose value of ECAP was above 4 and Negative, which includes nodes with ECAP value equal to or less than 4.

Considering this classification in both the ground truth and in the prediction sets, a confusion matrix was created. In it, the nodes that were classified as Positive in ground truth and prediction sets were labeled as True Positives (TP), the nodes that were classified as Negative in both sets were labeled as True Negatives (TN), the nodes that were erroneously classified as Negative in prediction set were labeled as False Negatives (FN) and the nodes which were incorrectly predicted as Positive, actually being Negative, were labelled as False Positive (FP).

With that classification in mind, four more metrics were computed: True Positive Rate (TPR), True Negative Rate (TNR) accuracy and F₁-score [36, 37]. TPR, which is also known as sensitivity, recall or hit rate, represents the proportion of nodes that were correctly catalogued as positive to the total number of nodes predicted as positive (Equation 6). On the other hand, the TNR, also called specificity or inverse recall, measures the proportion of nodes correctly identified as negative with respect to the total number of nodes predicted as negative (Equation 7). With respect to the accuracy, it measures the closeness of the predictions to the ground truth value, without mattering if they are positive or negative (Equation 8). Finally, the F₁-score, which is also called F-measure, represents the harmonic mean of precision (which is the proportion of correctly identified positive samples to the total positive predicted samples) and TPR (Equation 9). Basically, it gives more importance to present a low number of FP and FN labelled nodes.

$$TPR = \frac{TP}{TP + FN} \quad [6]$$

$$TNR = \frac{TN}{TN + FP} \quad [7]$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad [8]$$

$$F_1 - score = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad [9]$$

3. RESULTS

3.1. Flattening algorithm

The proposed flattening algorithm has been evaluated on 206 LAA geometries. The average time the algorithm takes to generate the whole dataset of flat images is 254.91 seconds, therefore, averaging, the elapsed time in generating one representation was 1.24 seconds.

Some examples of comparing 3D mesh and 2D flat representation can be seen in Figures 17 and 18. Note the difficulty to compare different 3D representations among them due to their different shape.

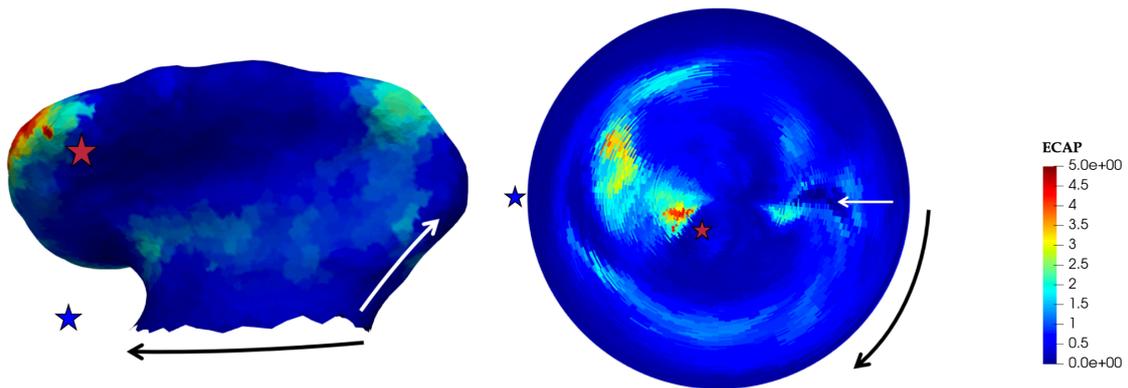


Figure 17. 3D structure and flat representation of a LAA achieved in this study. Direction of radial and angular mapping is displayed, in white and black, respectively. The blue star represents the relative position of the circumflex artery with respect to the LAA in both images. The brown star marks the same point in both representations. ECAP: Endothelial Cell Activation Potential.

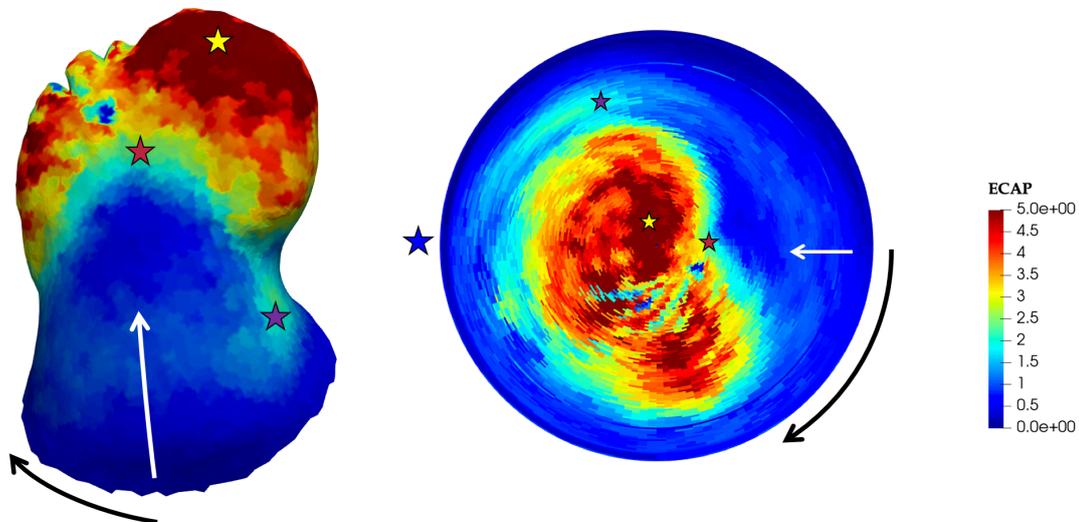


Figure 18. 3D structure and flat representation of a different LAA achieved in this study. Direction of radial and angular mapping is displayed, in white and black, respectively. The blue star represents the relative position of the circumflex artery with respect to the LAA, it is occluded by the LAA in the left image. The brown, yellow and violet stars mark point correspondences between both representations. ECAP: Endothelial Cell Activation Potential.

3.2. Sensitivity analysis

With respect to the visualization, as stated before, some input parameters of the flattening algorithm can be changed. These are the number of radial isolines, the number of angular isolines and the timestep. Firstly, the result after changing the number of radial isolines is shown in Figure 19 a); there, the output of the algorithm using 25, 50, 100 and 200 radial isolines is compared, keeping the number of angular isolines in 50. In Figure 19 b), it is displayed the output with the number of radial isolines set in 50, while the number of angular isolines varies from 25 to 50, 100 and finally, 200.

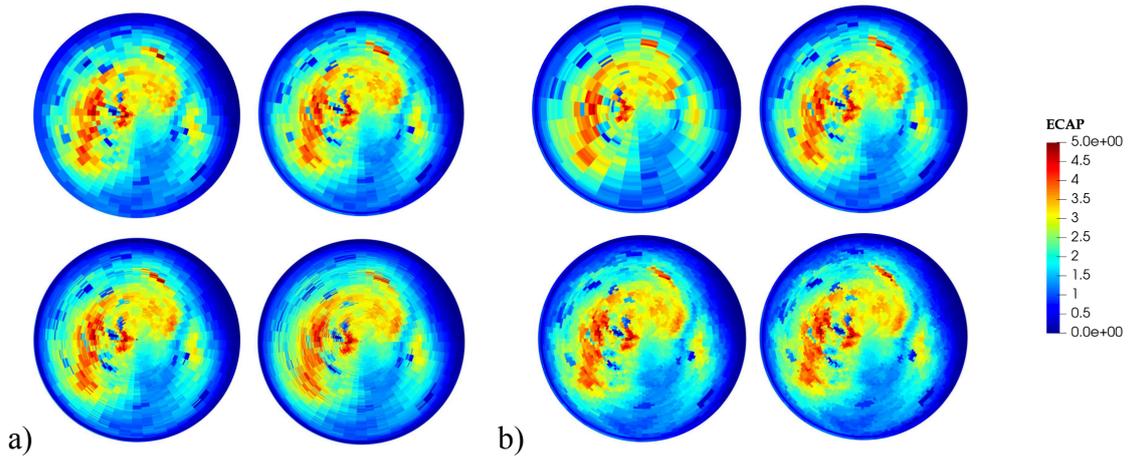


Figure 19. In (a), from left to right and from top to bottom: output of the algorithm with 25, 50, 100 and 200 radial isolines, respectively. The number of angular isolines is kept in 50. In (b), from left to right and from top to bottom: output of the algorithm with 25, 50, 100 and 200 angular isolines, respectively. The number of radial isolines is kept in 50. ECAP: Endothelial Cell Activation Parameter.

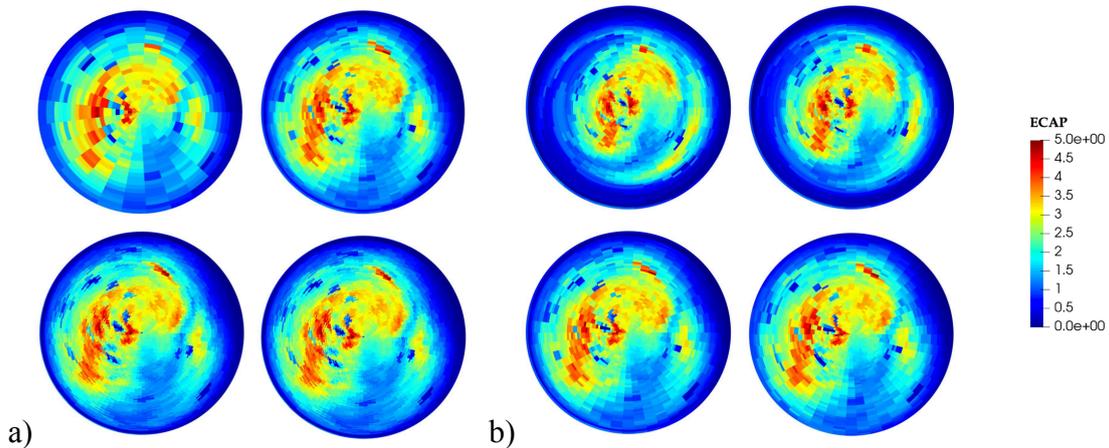


Figure 20. In (a), from left to right and from top to bottom: output of the algorithm with 25, 50, 100 and 200 radial and angular isolines. In (b), from left to right and from top to bottom: output of the algorithm with timesteps= 10^{-2} , 10^{-4} , 10^{-6} and 10^{-8} . The number of radial and angular isolines is 50. ECAP: Endothelial Cell Activation Parameter.

Finally, if radial and angular isolines vary at the same time, an increase or decrease of the resolution of the representation is achieved, as seen in Figure 20 a).

Another parameter that can be altered is the timestep. However, as it is explored in Figure 20 b), if it is changed outside the correct value, the representation becomes unreal. It is not simply a matter of resolution, but the problem is that nodes that do not exist are represented (left side of the bull's eye plot in cases with timestep equal to 10^{-2} and 10^{-4}).

3.3. Performance of the Deep Learning models

To assess the three proposed DL models, they were trained on a dataset composed of the 206 flat ECAP maps obtained from the flattening algorithm and the respective geometries. The selected input parameters for the algorithm were, in the case of the two DNN-based models: number of radial isolines=50, number of angular isolines=100 and timestep= 10^{-6} . In the case of the CNN-based model, the parameters were: number of radial isolines=65, number of angular isolines=65 and timestep= 10^{-6} .

The time that took each round of cross-validation in the DL training process for the DNN-based models was 1 minute and 42 seconds in the first one and 1 minute and 35 seconds in the second one, while in the CNN-based, the time that took the complete training was 9 minutes and 13 seconds. With respect to the metrics, the accuracy of the three approaches is compared in Table 1 among them and with the obtained in the study of Morales *et al.* [19]. The MAE and RMSE errors of the two DNN-based models were similar, around 0.73 and 0.97, respectively. In the CNN-based approach, those errors were 0.63 and 0.80.

Table 1. Errors of the predicted ECAP distributions for the three models and the one from Morales *et al.* [19] study. In the case of the DNN-based ones, after a 100-fold cross validation. Std stands for "Standard deviation".

	Liang model		Morales model		CNN-based model		Morales <i>et al.</i> study	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
MAE	0.744	0.29	0.727	0.06	0.627	0.10	0.646	0.05
RMSE	0.955	0.35	0.991	0.10	0.804	0.09	0.891	0.09

The predicted ECAP 2D mapping of a representative LAA of the three studied DL networks, together with the ground truth ECAP map are displayed in Figure 21. Furthermore, Figure 22 shows the prediction of the ECAP distribution on the LAA geometry by the three DL networks alongside with the prediction of Morales *et al.* [19] study and the ground truth.

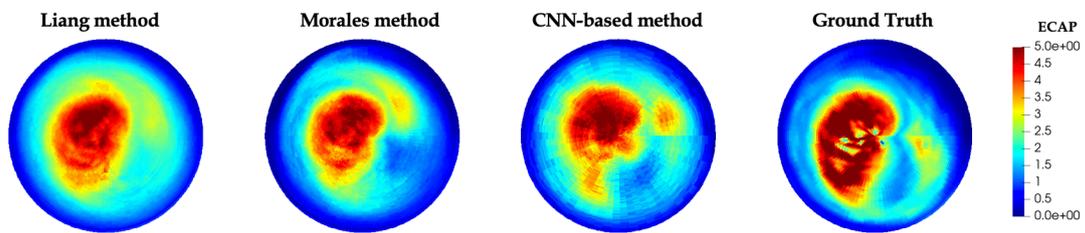


Figure 21. From left to right: predicted ECAP bull's eye plot mapping of Liang method, of Morales method, of the CNN-based method and ground truth. ECAP: Endothelial Cell Activation Parameter.

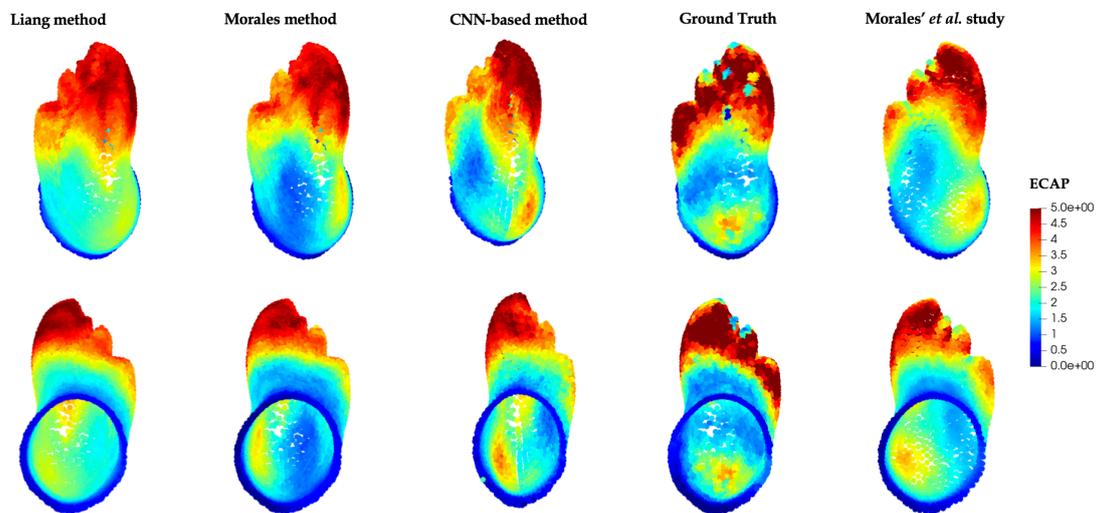


Figure 22. From left to right: predicted ECAP 3D mapping of Liang method, of Morales method, of the CNN-based method, ground truth ECAP distribution and prediction of Morales *et al.* study [19]. Top and bottom rows represent frontal and back views, respectively. ECAP: Endothelial Cell Activation Parameter.

Table 2 summarizes the average confusion matrices of the three approaches and compares them with the results of the study of Morales *et al.* [19] and Figure 23 shows the distribution of TP, TN, FP and FN nodes in the same LAA geometries than in Figure 21.

Table 2. Average number and percentage of nodes classified as FN (false negative), TN (true negative), TP (true positive) and FP (false positive) in the four methods. Total number of nodes: Liang and Morales models: 5000, CNN-based model: 4225, Morales study: 2466.

	Liang model	Morales model	CNN-based model	Morales <i>et al.</i> study
FN	570.1 (11.40%)	513.6 (10.27%)	255.2 (6.04%)	161.5 (6.55%)
TN	3573.8 (71.46%)	3554.1 (71.09%)	3057.9 (72.38%)	2023.7 (82.06%)
TP	476.4 (9.53%)	532.8 (10.66%)	653.9 (15.48%)	230.6 (9.35%)
FP	379.7 (7.59%)	399.5 (7.99%)	240.0 (5.68%)	50.2 (2.04%)

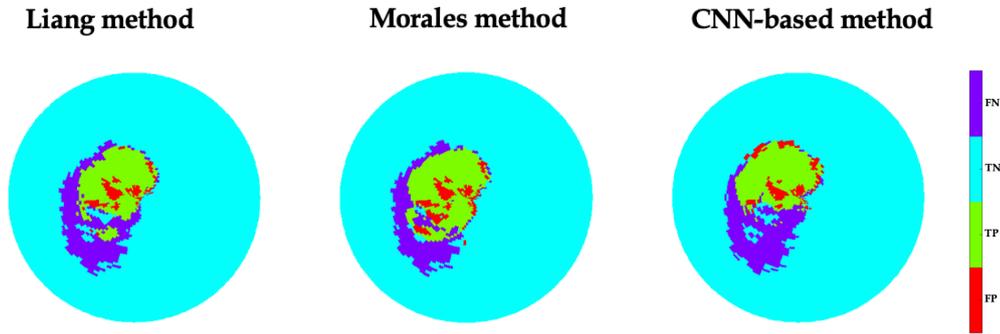


Figure 23. From left to right: predicted FN, TN, TP and FP areas in Liang method, in Morales method and in the CNN-based method. FN: False Negative, TN: True Negative, TP: True Positive, FP: False Positive.

Finally, TPR, TNR, accuracy and F_1 -score of the three proposed DL models and that of Morales *et al.* study [19] are shown in Table 3. In it, is shown that the CNN-based approach presents the higher TPR and F_1 -score, 0.72 and 0.73, respectively. On the other hand, in Morales *et al.* [19] study, the higher TNR and accuracy were achieved, 0.98 and 0.91, respectively.

Table 3. TPR, TNR, accuracy and F_1 -score of the three proposed DL models and the one from Morales et al. [19].

	Liang model	Morales model	CNN-based model	Morales <i>et al.</i> study
TNR	0.904 (90.4%)	0.889 (88.9%)	0.927 (92.7%)	0.976 (97.6%)
TPR	0.455 (45.5%)	0.509 (50.9%)	0.719 (71.9%)	0.582 (58.2%)
Accuracy	0.810 (81.0%)	0.817 (81.7%)	0.878 (87.9%)	0.914 (91.4%)
F_1-score	0.501 (50.1%)	0.539 (53.99%)	0.725 (72.5%)	0.685 (68.5%)

4. DISCUSSION

The basis of this work has been the generation of a fast surrogate of CFD simulations, based on CNN, to estimate the ECAP values of the LAA from its geometry. For that purpose, firstly a fast flattening algorithm of the LAA was generated from scratch. As previously detailed, this algorithm is able to perform the contour discretization and the flattening of a single LAA geometry in a little over a second.

By using this method, we successfully managed to flatten the LAA allowing employ standard state of the art DL algorithms which require Euclidean training datasets, without the need of performing any remeshing of the LAA mesh before the CFD simulations. In addition, correspondences among different LAAs are achieved in an easier and faster way than in other approaches, e.g. template registration. It is important to emphasize the advantage of being able to perform the flattening of a LAA geometry practically instantaneously, allowing to obtain results in real time.

Moreover, the proposed flattening allows for quick visual examination of larger LAA datasets. Since LAA presents a complex anatomy, the distribution of the ECAP on its surface is currently interpreted by rotating the LAA in the interface of a computer visualization application. Although this method is well suited for in-depth analysis of one LAA at a time, this task becomes daunting in the case of comparing large datasets. Hence, one way to overcome this problem is by using the proposed flattening framework. In addition, we hypothesize that this algorithm might be easily applied to other topologically similar structures, since it is only necessary to have a starting edge and a reference point to implement it.

With respect to the DL models evaluated in this study, they have been able to correctly predict ECAP distribution in LAA surface but presenting difficulties in some cases. As seen in the Results section, the MAE and the RMSE are similar between the two DNN-based models, similarly to the accuracy and the F₁-score. These metrics suggest that, by the moment, the non-rigid volumetric registration method used by Slipsager *et al.* [26] performs better using DNN-based surrogate models than the contour discretization approach based on the flattening algorithm. However, the method proposed in this work is orders of magnitude faster than the other one. On the other hand, the CNN-based surrogate model has lower MAE and RMSE values than the DNN-based ones, including the two analyzed in this study and the one studied by Morales *et al.* [19]. It has a better

performance than the two DNN-based methods examined in this paper on each metric analyzed and presents a better TPR and F₁-score than the method proposed by Morales *et al.* [19]. It suggests that the CNN-based model is able to predict the areas with a higher value of ECAP with more precision than the other approach, while the TNR does not vary much.

A problem seen in the Results section is that, although the TNR is around 90% in the three studied models, the global accuracy decreases due to the lower TPR. This problem is especially important in the DNN-based models, which do not predict well the areas with high values of ECAP and gets confused in some cases. This problem may happen due to the limited size of the used dataset, that do not contain a significative amount of geometries. It leads to failure in the non-linear mapping step of the DL networks, because they have not learnt what is the ECAP distribution map of geometries such that and the network must forecast the ECAP of the nodes computing it from not enough similar geometries. Nevertheless, it must not be forgotten that the task in hand involves a high degree of difficulty and that the DL networks used have shown a satisfactory predictive capability in most cases.

A further point to note is that the selection of 4 as threshold for dividing the classifying the nodes into two categories has been performed without existing validated ECAP maps on the LAA in the literature [8]. If other value were selected, the accuracy of the models may change significantly.

Furthermore, it is worth mentioning that the LA modelling as an ovoid without any inter-patient difference produces unrealistic ECAP maps, which cannot be considered clinically relevant at the moment. A method that could be attempted to bridge this problem would consist on obtaining a big enough dataset of real LAs. Correspondences among them could be set up by means of the fast flattening method proposed by Núñez-García *et al.* [23].

Another observed limitation is that, at this time, networks are likely to fail to predict the distribution of ECAP if they are tested on real geometries, due to the limited size of the dataset in which the original SSM was based and because of the high inter-patient variability that the LAA presents. With this in mind, the approach to overcome this problem would consist on using a larger dataset of real patients to create the SSM. Hence, the model will be more precise. However, it is important to bear in mind that obtaining

clinical information and putting it together, combining data from different sources, is a slow and challenging process filled with, inter alia, bureaucracy and permit applications. An additional limitation of this study is the possible low accuracy of the ECAP maps that were used as input for the proposed flattening and DL algorithms. The pre-processing performed by Morales *et al.* may have altered the geometry of the LAA to the point of influencing the subsequent CFD simulations. In particular, the remeshing process of the LAAs, having to preserve the correspondences among nodes, results in areas of the LAA in which the blood cannot flow in the correct way, in CFD simulations. It could affect the computed ECAP values, thus altering the DL model. Using the flattening technique to establish correspondences, there will not be need for remeshing the LAA in such an unnatural manner, thus giving as a result more precise ECAP maps to train the DL models better.

5. CONCLUSIONS AND FUTURE DIRECTIONS

It is important to bear in mind that LAA is a small part of the human heart and that the quantitative obtained error in the ECAP distribution prediction may seem extremely important when looking it in the computer, but actually, it only supposes a difference of millimeters in the human body. In this study, most of the predicted ECAP maps were similar qualitatively to the ground truth; there were few cases in which they differed so much that they could lead to misdiagnosis.

However, apart from the commented limitations of this study, it is true that the use of DL in clinical world is, not only a promising approach, but a technique used nowadays in hospitals worldwide. Using DL, patient-specific data can be produced in matter of seconds and hence DL linked with CFD enables using more crucial data in real time. It is important in more areas than only cardiology, which are the design of medical devices or planning surgeries.

As future step, since flattening has shown its remarkable potential, that algorithm could be used to create a wide dataset of flat ECAP maps to be later used to train a new DL model. The input ECAP maps to the algorithm could be obtained without geometric constraints, allowing more realistic results and the diversity of LAA geometries would be larger, allowing the model to capture better the non-linear relationship between the anatomic features and the ECAP.

1. LIST OF FIGURES

	Page.
Figure 1. Watchman™ and Amplatzer™ Cardiac Plug devices after implantation in the LAA. From [4].	1
Figure 2. Endocasts obtained from explanted hearts showing the different LAA intraluminal morphologies. From [6].	2
Figure 3. Flat representation of the aorta (a), from [24] and Cardiac Bull's Eye Plot (b), from [25].	5
Figure 4. General pipeline of the study. The four steps related to geometry and ECAP obtaining were performed by Morales <i>et al.</i> [19]. Following the enumeration, the LAA are remeshed (1) and introduced into the flattening algorithm, where a radial (2) and an angular (3) mappings are applied over them, to obtain the flat representation (4). Afterwards, the DL networks are trained to estimate the ECAP in new LAA geometries. LAA: Left Atrial Appendage, SSM: Statistical Shape Model, CFD: Computational Fluid Dynamics, ECAP: Endothelial Cell Activation Potential.	6
Figure 5. Comparison of geometries before and after performing the remeshing. In the left, original LAA; in the right, manually remeshed LAA. ECAP: Endothelial Cell Activation Potential.	7
Figure 6. A schematic for illustrating the calculation of polar-based coordinates. From left to right: 3D mesh, which is the input of the algorithm; radial (top) and angular (bottom) isolines obtained from radial and angular mapping, respectively, flat representation of the cartesian grid and corresponding nodes in the 3D geometry. Adapted from Meuschke et al. [28].	8
Figure 7. Isolines of the LAA. (a) 10 isolines, (b) 30 isolines. In black, the nodes of the LAA. Colored, the nodes corresponding to the isolines.	9
Figure 8. Nodes of the isolines before (a) and after (b) being projected into their respective 3D planes.	10
Figure 9. Graphic representation of the angular (a) and radial (b) mapping. In (a), nodes are colored according to the angular distance to the reference node. The white arrow crosses the different nodes of a radial isoline, with increasing angular distance values. In (b), nodes are colored according to the distance to the ostium. The white arrow crosses the different nodes of an angular isoline, with increasing distance to the ostium. N is the reference node of each radial isoline, diametrically opposite the circumflex artery.	11

Figure 10. 3D structure and flat representation of the LAA achieved in this study. Direction of radial and angular mapping is displayed, in red and green, respectively. The apex and a point of the ostium are emphasized. The blue star represents the relative position of the circumflex artery with respect to the LAA. ECAP: Endothelial Cell Activation Potential. 12

Figure 11. Overview of the DNN workflow, adapted from Liang *et al.* [18]. The input to the network is the shape of the LAA, represented by the coordinates of the nodes and the output is the ECAP distribution in each node. α_{1-M} and β_{1-N} are the components of the shape code and of the ECAP code of the LAA, respectively. 13

Figure 12. Network used for the shape encoding by means of PCA. x_i , y_i and z_i are the coordinates of the node $i \in (1,5000)$ and α_{1-20} are the components of the shape code of the LAA. 14

Figure 13. Fully connected neural network used for the non-linear mapping. α_{1-20} and β_{1-64} are the components of the shape code and of the ECAP code, of the LAA, respectively 14

Figure 14. Network used for the ECAP decoding. β_{1-64} are the components of the ECAP code of the LAA, r_{1-64} are the components of the ECAP code of each region, from (1) to (25) and $E^{(1)}$ to $E^{(25)}$ are the values of ECAP of each region. 15

Figure 15. Workflow of the second DL model. The input consisted on the coordinates of the nodes, shape encoding and decoding steps were performed by truncated PCA and the output consisted on the ECAP values of the nodes. 15

Figure 16. Overall encoder-decoder CNN architecture. Given the features from the encoder part (red arrows), the ECAP prediction was obtained through multiple decoding operations (blue arrows). 17

Figure 17. 3D structure and flat representation of a LAA achieved in this study. Direction of radial and angular mapping is displayed, in white and black, respectively. The blue star represents the relative position of the circumflex artery with respect to the LAA in both images. The maroon star marks the same point in both representations. ECAP: Endothelial Cell Activation Potential. 20

Figure 18. 3D structure and flat representation of a different LAA achieved in this study. Direction of radial and angular mapping is displayed, in white and black, respectively. The blue star represents the relative position of the circumflex artery with respect to the LAA, it is occluded by the LAA in the left image. The maroon, yellow and violet stars mark point correspondences between both representations. ECAP: Endothelial Cell Activation Potential. 20

Figure 19. In (a), from left to right and from top to bottom: output of the algorithm with 25, 50, 100 and 200 radial isolines, respectively. The number of angular isolines is kept in 50. In (b), from left to right and from top to bottom: output of the algorithm with 25, 50, 100 and 200 angular isolines, respectively. The number of radial isolines is kept in 50. ECAP: Endothelial Cell Activation Parameter.	21
Figure 20. In (a), from left to right and from top to bottom: output of the algorithm with 25, 50, 100 and 200 radial and angular isolines. In (b), from left to right and from top to bottom: output of the algorithm with timesteps= 10^{-2} , 10^{-4} , 10^{-6} and 10^{-8} . The number of radial and angular isolines is 50. ECAP: Endothelial Cell Activation Parameter.	21
Figure 21. From left to right: predicted ECAP bull's eye plot mapping of Liang method, of Morales method, of the CNN-based method and ground truth. ECAP: Endothelial Cell Activation Parameter.	23
Figure 22. From left to right: predicted ECAP 3D mapping of Liang method, of Morales method, of the CNN-based method, ground truth ECAP distribution and prediction of Morales <i>et al.</i> study [19]. Top and bottom rows represent frontal and back views, respectively. ECAP: Endothelial Cell Activation Parameter.	23
Figure 23. From left to right: predicted FN, TN, TP and FP areas in Liang method, in Morales method and in the CNN-based method. FN: False Negative, TN: True Negative, TP: True Positive, FP: False Positive.	24
Figure 25. Set of predictions and comparison to the ground truth value of the first LAA of a set of six. In order: bull's eye plot representation, frontal and back views of 3D representation and bull's eye plot representation of TP, FP, TN and FN.	39
Figure 26. Set of predictions and comparison to the ground truth value of the second LAA of a set of six. In order: bull's eye plot representation, 2D square representation, frontal and back views of 3D representation and bull's eye plot and 2D square representation of TP, FP, TN and FN.	40
Figure 27. Set of predictions and comparison to the ground truth value of the third LAA of a set of six. In order: bull's eye plot representation, frontal and back views of 3D representation and bull's eye plot representation of TP, FP, TN and FN.	41
Figure 28. Set of predictions and comparison to the ground truth value of the fourth LAA of a set of six. In order: bull's eye plot representation, frontal and back views of 3D representation and bull's eye plot representation of TP, FP, TN and FN.	42

Figure 29. Set of predictions and comparison to the ground truth value of the fifth LAA of a set of six. In order: bull's eye plot representation, frontal and back views of 3D representation and bull's eye plot representation of TP, FP, TN and FN. 43

Figure 30. Set of predictions and comparison to the ground truth value of the sixth LAA of a set of six. In order: bull's eye plot representation, frontal and back views of 3D representation and bull's eye plot representation of TP, FP, TN and FN. 44

2. LIST OF TABLES

	Page
Table 1. Errors of the predicted ECAP distributions for the three models and the one from Morales <i>et al.</i> [19] study. In the case of the DNN-based ones, after a 100-fold cross validation. Std stands for “Standard deviation”.	22
Table 2. Average number and percentage of nodes classified as FN (false negative), TN (true negative), TP (true positive) and FP (false positive) in the four methods. Total number of nodes: Liang and Morales models: 5000, CNN-based model: 4225, Morales study: 2466.	24
Table 3. TPR, TNR, accuracy and F ₁ -score of the three proposed DL models and the one from Morales et al. [19].	25

3. BIBLIOGRAPHY

- [1] A. Hersi, L. B. Mitchell, and D. G. Wyse, “Atrial fibrillation: Challenges and opportunities,” *Can. J. Cardiol.*, vol. 22, no. SUPPL. C, pp. 21C–26C, 2006, doi: 10.1016/s0828-282x(06)70998-9.
- [2] A. S. Go *et al.*, “Prevalence of Diagnosed Atrial Fibrillation in Adults,” *Jama*, vol. 285, no. 18, p. 2370, 2001, doi: 10.1001/jama.285.18.2370.
- [3] “What is Atrial Fibrillation (AFib or AF)?,” *Heart.org*. [Online]. Available: <https://www.heart.org/en/health-topics/atrial-fibrillation/what-is-atrial-fibrillation-afib-or-af>.
- [4] X. Freixa and V. Martín-Yuste, “Percutaneous left atrial appendage occlusion in 2016,” *Revista Española de Cardiología*, vol. 66, no. 12, pp. 919–922, 2013.
- [5] A. Cresti *et al.*, “Prevalence of extra-appendage thrombosis in non-valvular atrial fibrillation and atrial flutter in patients undergoing cardioversion: a large transoesophageal echo study,” *EuroIntervention*, vol. 15, no. 3, pp. e225–e230, 2019, doi: 10.4244/eij-d-19-00128.
- [6] R. Beigel, N. C. Wunderlich, S. Y. Ho, R. Arsanjani, and R. J. Siegel, “The left atrial appendage: Anatomy, function, and noninvasive evaluation,” *JACC Cardiovasc. Imaging*, vol. 7, no. 12, pp. 1251–1265, 2014, doi: 10.1016/j.jcmg.2014.08.009.
- [7] A. Masci *et al.*, “The impact of left atrium appendage morphology on stroke risk assessment in atrial fibrillation: A computational fluid dynamics study,” *Front. Physiol.*, vol. 9, no. January, pp. 1–11, 2019, doi: 10.3389/fphys.2018.01938.
- [8] P. Di Achille, G. Tellides, C. A. Figueroa, and J. D. Humphrey, “A haemodynamic predictor of intraluminal thrombus formation in abdominal aortic aneurysms,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 470, no. 2172, 2014, doi: 10.1098/rspa.2014.0163.
- [9] G. García-Isla *et al.*, “Sensitivity analysis of geometrical parameters to study haemodynamics and thrombus formation in the left atrial appendage,” *Int. j. numer. method. biomed. eng.*, vol. 34, no. 8, 2018, doi: 10.1002/cnm.3100.
- [10] L. Wu *et al.*, “Relation of Left Atrial Appendage Morphology Determined by

- Computed Tomography to Prior Stroke or to Increased Risk of Stroke in Patients With Atrial Fibrillation,” *Am. J. Cardiol.*, vol. 123, no. 8, pp. 1283–1286, 2019, doi: 10.1016/j.amjcard.2019.01.024.
- [11] M. García-Villalba *et al.*, “Demonstration of Patient-Specific Simulations To Assess Left Atrial Appendage Thrombogenesis Risk,” 2020.
- [12] G. M. Bosi *et al.*, “Computational Fluid Dynamic Analysis of the Left Atrial Appendage to Predict Thrombosis Risk,” *Front. Cardiovasc. Med.*, vol. 5, no. April, pp. 1–8, 2018, doi: 10.3389/fcvm.2018.00034.
- [13] A. Masci *et al.*, “Development of a computational fluid dynamics model of the left atrium in atrial fibrillation on a patient specific basis,” *Comput. Cardiol. (2010).*, vol. 44, pp. 1–4, 2017, doi: 10.22489/CinC.2017.004-429.
- [14] G. Grigoriadis *et al.*, “Computational Fluid Dynamics of Blood Flow at the Left Atrium and Left Atrium Appendage,” in *IFMBE Proceedings*, 2019, vol. 76, pp. 938–946, doi: 10.1007/978-3-030-31635-8_232.
- [15] “Ascent of machine learning in medicine,” *Nat. Mater.*, vol. 18, no. 5, p. 407, 2019, doi: 10.1038/s41563-019-0360-1.
- [16] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, “Machine Learning for Fluid Mechanics,” *Annu. Rev. Fluid Mech.*, vol. 52, no. 1, pp. 477–508, 2020, doi: 10.1146/annurev-fluid-010719-060214.
- [17] J. N. Kutz, “Deep learning in fluid dynamics,” *J. Fluid Mech.*, vol. 814, pp. 1–4, 2017, doi: 10.1017/jfm.2016.803.
- [18] L. Liang, M. Liu, C. Martin, and W. Sun, “A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis,” *J. R. Soc. Interface*, vol. 15, no. 138, 2018, doi: 10.1098/rsif.2017.0844.
- [19] X. Morales, J. Mill, K. A. Juhl, and A. Olivares, “Deep Learning Surrogate of Computational Fluid Dynamics for Thrombus Formation Risk in the Left Atrial Appendage,” in *Statistical Atlases and Computational Models of the Heart. Multi-Sequence CMR Segmentation, CRT-EPiggy and LV Full Quantification Challenges*, 2020, vol. 7, pp. 1–9.
- [20] S. Saha, “A Comprehensive Guide to Convolutional Neural Networks — the

- ELI5 way,” *towards data science*, 2018. [Online]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [21] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, “Medical Image Analysis using Convolutional Neural Networks: A Review,” *J. Med. Syst.*, vol. 42, no. 11, 2018, doi: 10.1007/s10916-018-1088-1.
- [22] J. Kreiser, M. Meuschke, G. Mistelbauer, B. Preim, and T. Ropinski, “A Survey of Flattening-Based Medical Visualization Techniques,” *Comput. Graph. Forum*, vol. 37, no. 3, pp. 597–624, 2018, doi: 10.1111/cgf.13445.
- [23] M. Nunez-Garcia *et al.*, “Fast quasi-conformal regional flattening of the left atrium,” *IEEE Trans. Vis. Comput. Graph.*, pp. 1–1, 2020, doi: 10.1109/tvcg.2020.2966702.
- [24] B. Köhler, M. Meuschke, U. Preim, K. Fischbach, M. Gutberlet, and B. Preim, “2D plot visualization of aortic vortex flow in cardiac 4D PC-MRI data,” *Inform. aktuell*, pp. 257–262, 2015, doi: 10.1007/978-3-662-46224-9_45.
- [25] M. D. Cerqueira *et al.*, “Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart: A Statement for Healthcare Professionals from the Cardiac Imaging Committee of the Council on Clinical Cardiology of the American Heart Association,” *Circulation*, vol. 105, no. 4, pp. 539–542, 2002, doi: 10.1161/hc0402.102975.
- [26] J. M. . et al. Slipsager, “Statistical Shape Clustering of Left Atrial Appendages,” Springer International Publishing, 2019.
- [27] J. Mill, A. L. Olivares, J. Noailly, X. Freixa, and Ó. Cámara Rey, “Optimal boundary conditions in fluid simulations for predicting occlude related thrombus formation in the left atria,” in *Computational and Mathematical Biomedical Engineering, Sixth International Conference*, 2019.
- [28] M. Meuschke, S. Oeltze-Jafra, O. Beuing, B. Preim, and K. Lawonn, “Classification of Blood Flow Patterns in Cerebral Aneurysms,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 7. pp. 2404–2418, 2019, doi: 10.1109/TVCG.2018.2834923.

- [29] K. Crane, C. Weischedel, and M. Wardetzky, “Geodesics in heat: A new approach to computing distance based on heat flow,” *ACM Trans. Graph.*, vol. 32, no. 5, 2013, doi: 10.1145/2516971.2516977.
- [30] T. Heimann and H. P. Meinzer, “Statistical shape models for 3D medical image segmentation: A review,” *Med. Image Anal.*, vol. 13, no. 4, pp. 543–563, 2009, doi: 10.1016/j.media.2009.05.004.
- [31] H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li, “Improving deep neural networks using softplus units,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2015-Septe, pp. 0–3, 2015, doi: 10.1109/IJCNN.2015.7280459.
- [32] I. Markovskiy, “Structured low-rank approximation and its applications,” *Automatica*, vol. 44, no. 4, pp. 891–909, 2008, doi: 10.1016/j.automatica.2007.09.011.
- [33] A. Li, R. Chen, A. B. Farimani, and Y. J. Zhang, “Reaction diffusion system prediction based on convolutional neural network,” *Sci. Rep.*, vol. 10, no. 1, pp. 1–9, 2020, doi: 10.1038/s41598-020-60853-2.
- [34] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017, doi: 10.1109/TPAMI.2016.2644615.
- [35] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.
- [36] A. Tharwat, “Classification assessment methods,” *Appl. Comput. Informatics*, 2018, doi: 10.1016/j.aci.2018.08.003.
- [37] M. Sokolova, N. Japkowicz, and S. Szpakowicz, “Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation,” *AAAI Work. - Tech. Rep.*, vol. WS-06-06, pp. 24–29, 2006, doi: 10.1007/11941439_114.

4. APPENDIX

4.1. Appendix: Performance results of the DL networks.

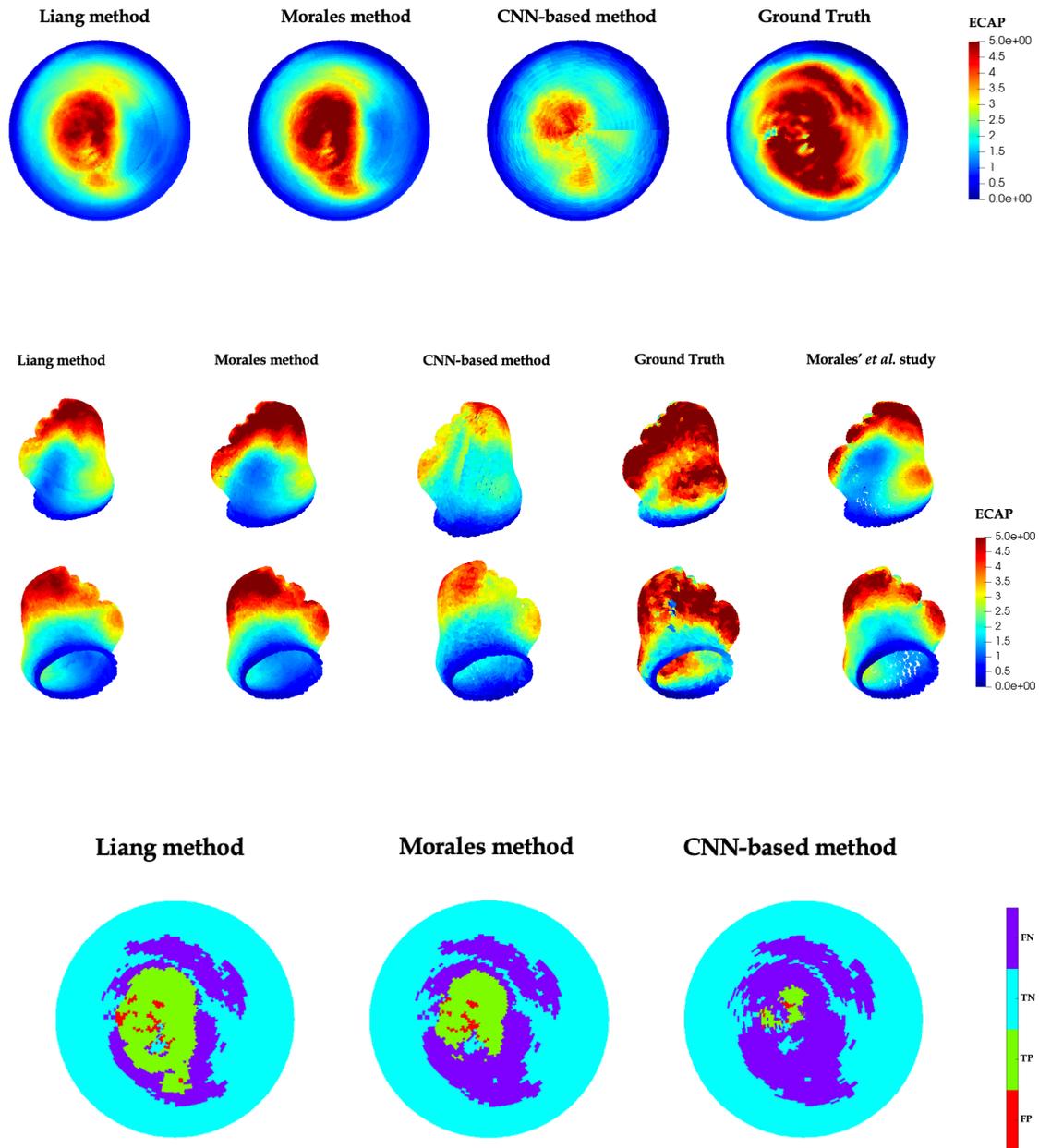


Figure 24. Set of predictions and comparison to the ground truth value of the first LAA of a set of six. In order: bull's eye plot representation, frontal and back views of 3D representation and bull's eye plot representation of TP, TN and FN.

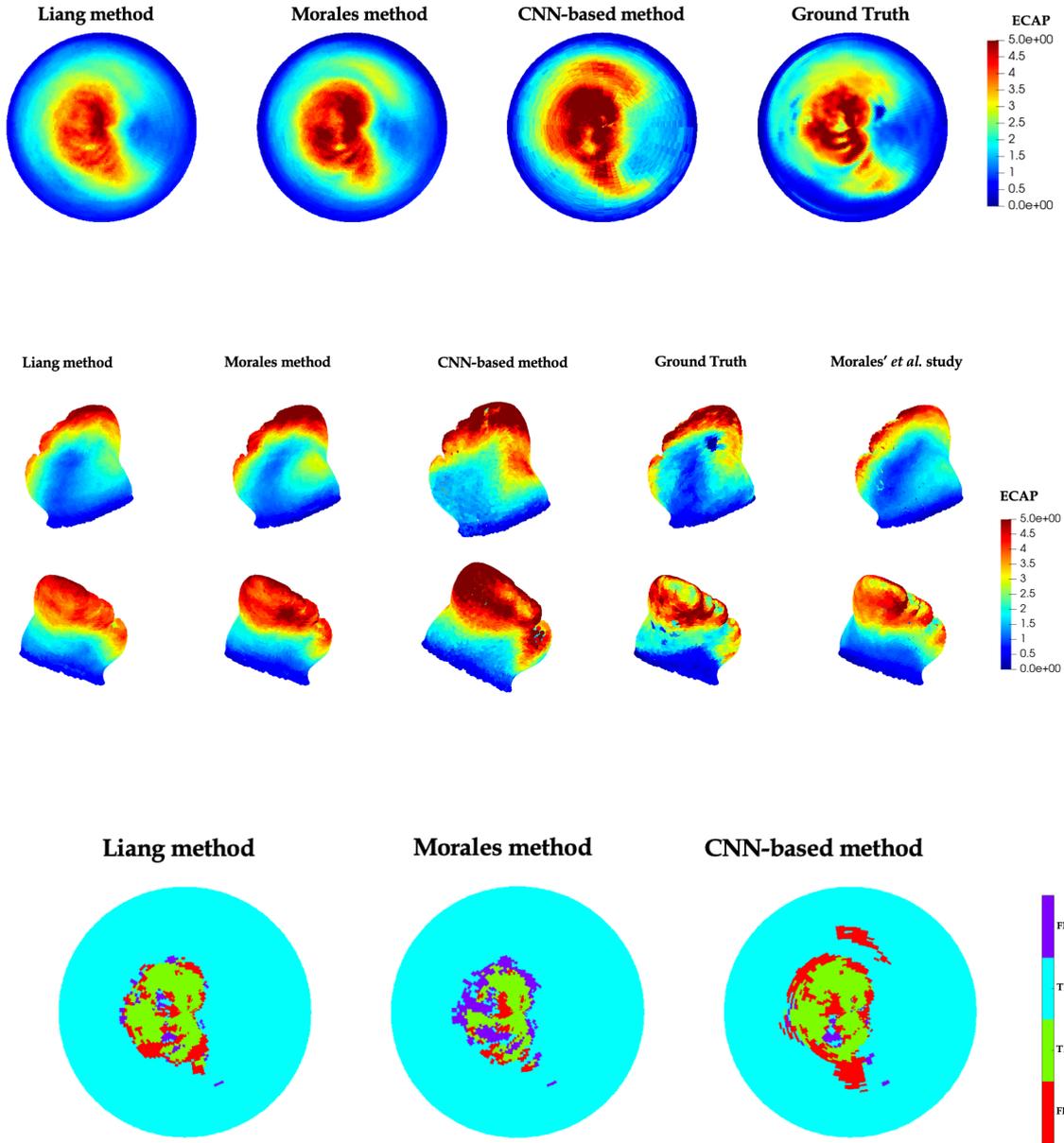


Figure 25. Set of predictions and comparison to the ground truth value of the second LAA of a set of six. In order: bull's eye plot representation, frontal and back views of 3D representation and bull's eye plot representation of TP, FP, TN and FN.

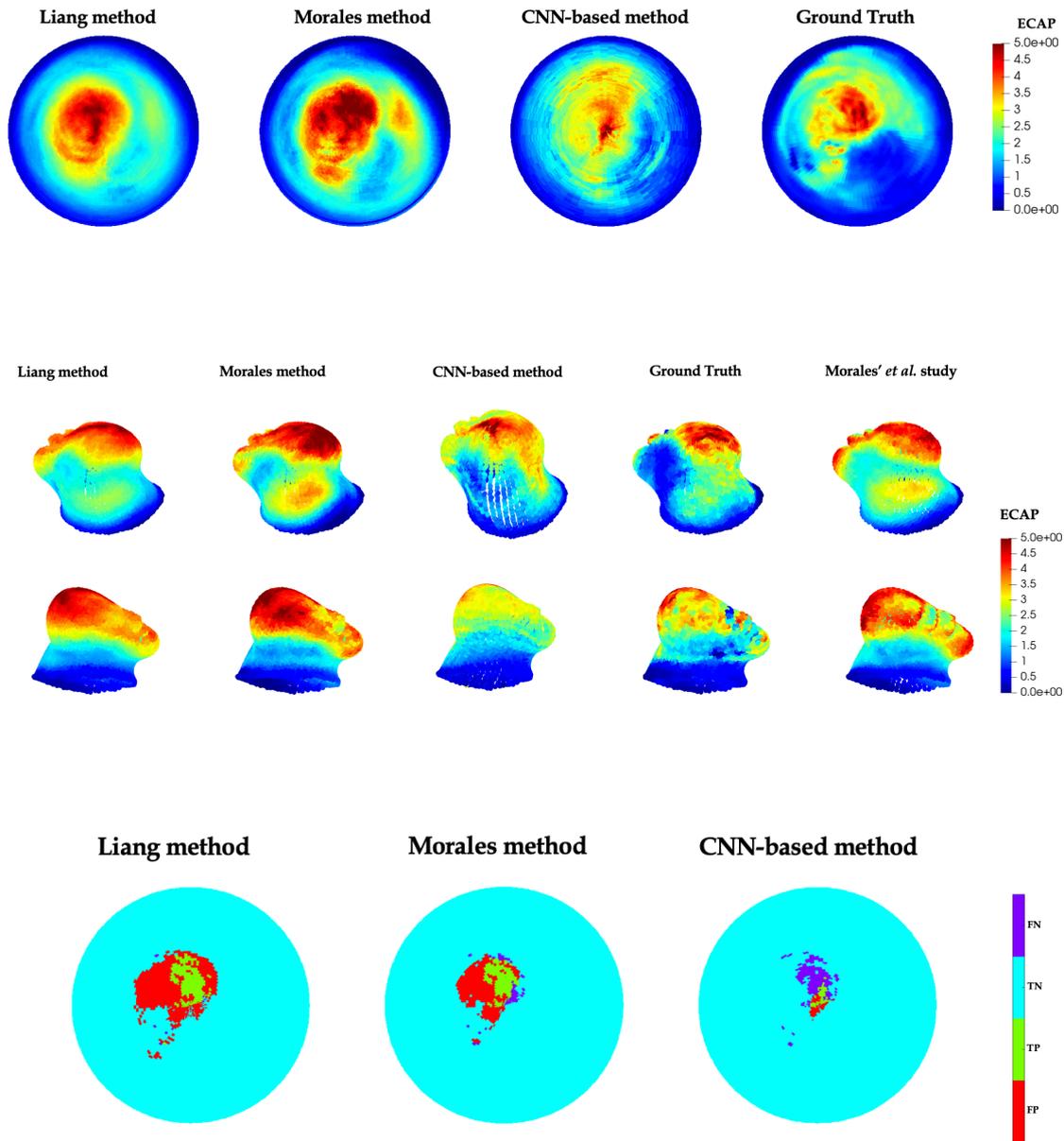


Figure 26. Set of predictions and comparison to the ground truth value of the third LAA of a set of six. In order: bull's eye plot representation, frontal and back views of 3D representation and bull's eye plot representation of TP, FP, TN and FN.

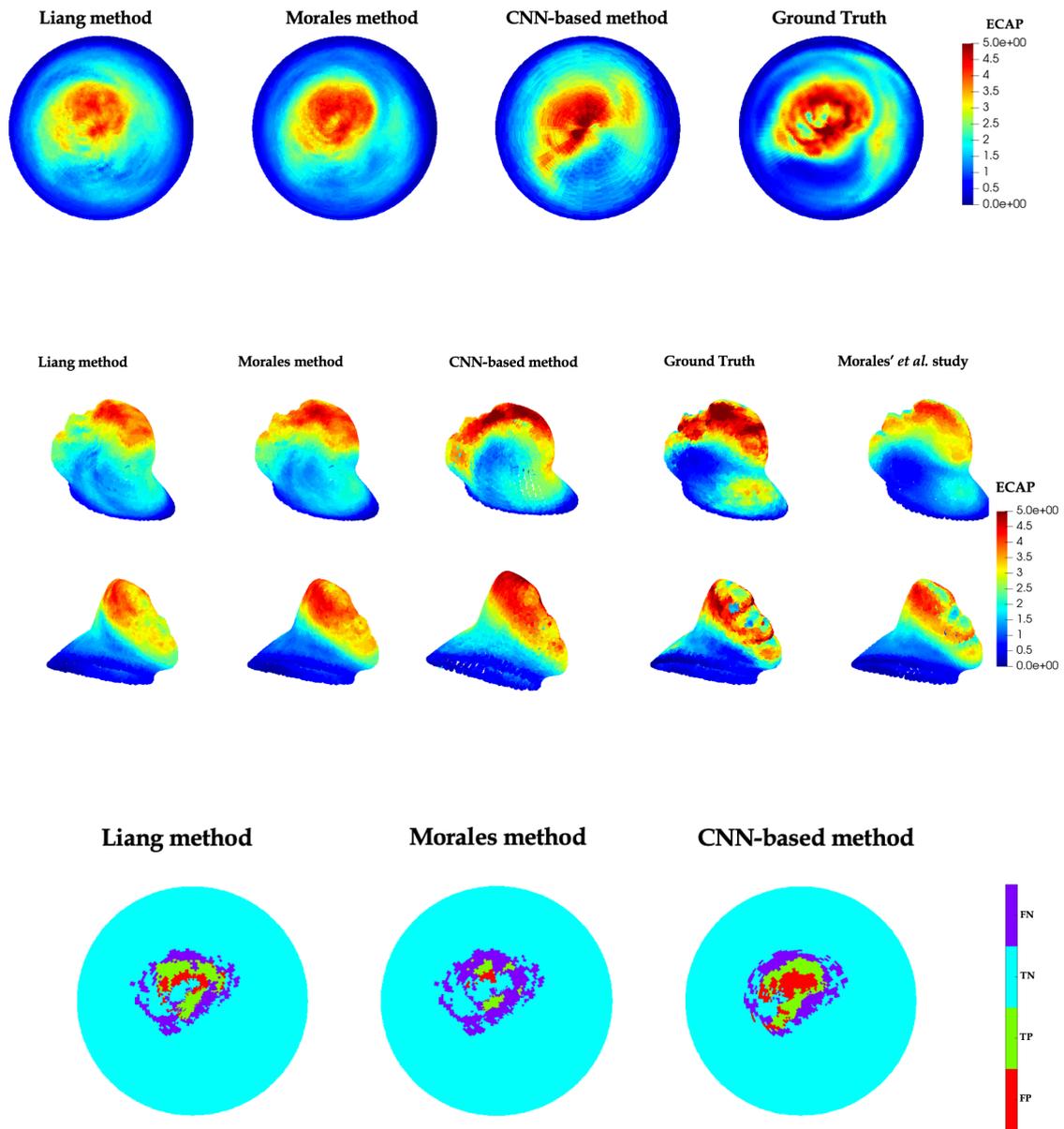


Figure 27. Set of predictions and comparison to the ground truth value of the fourth LAA of a set of six. In order: bull's eye plot representation, frontal and back views of 3D representation and bull's eye plot representation of TP, FP, TN and FN.

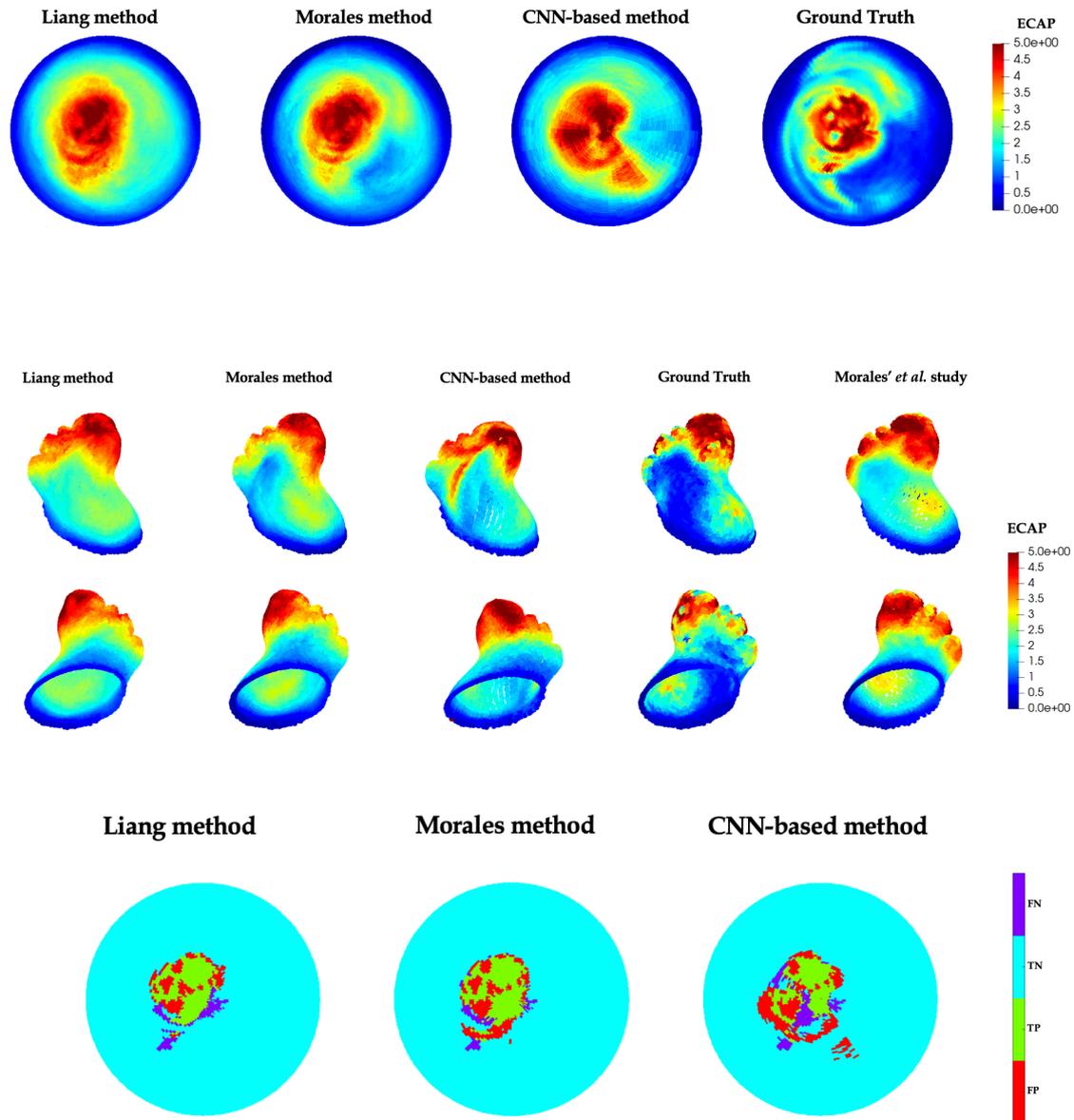


Figure 28. Set of predictions and comparison to the ground truth value of the fifth LAA of a set of six. In order: bull's eye plot representation, frontal and back views of 3D representation and bull's eye plot representation of TP, FP, TN and FN.

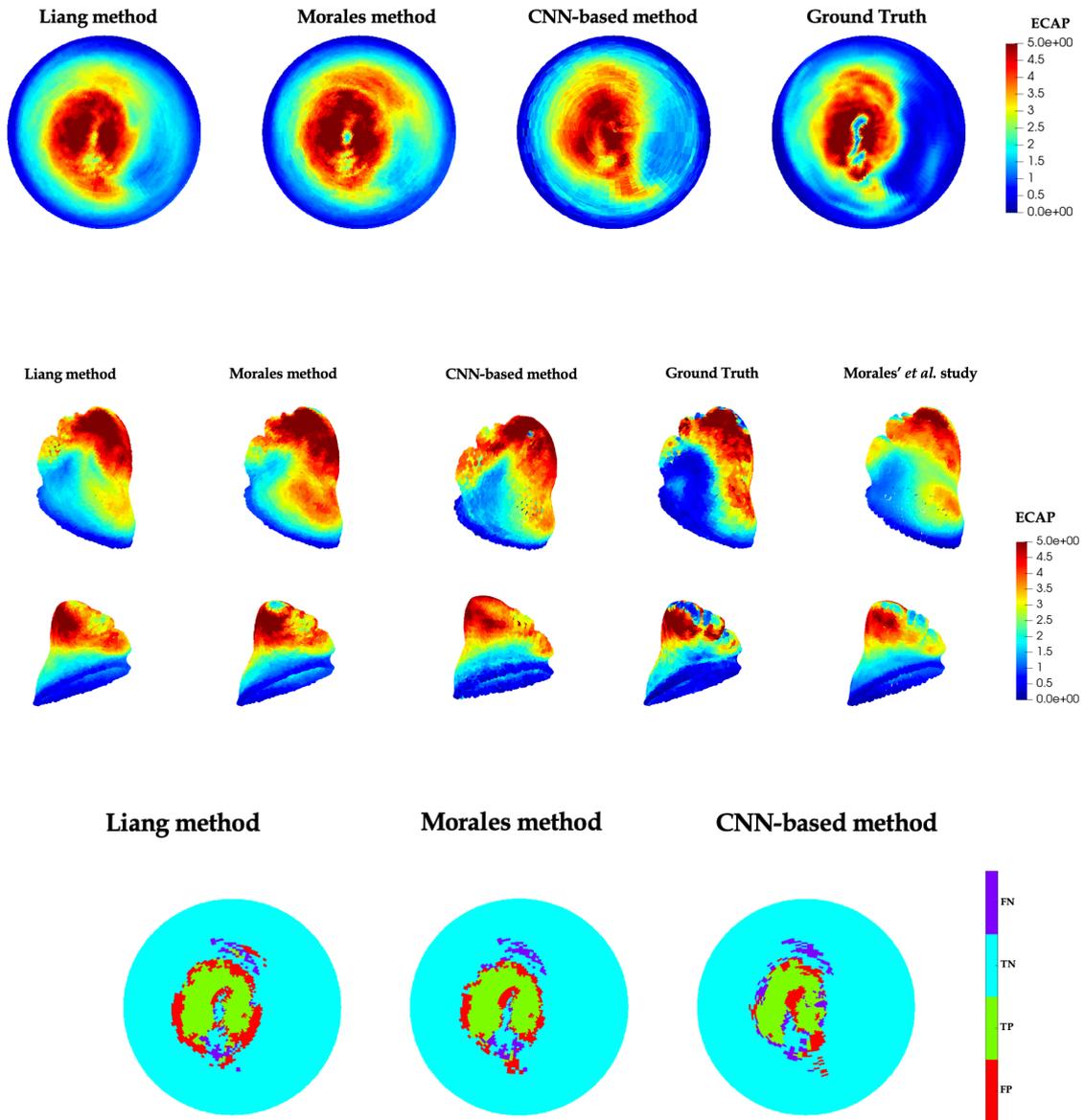


Figure 29. Set of predictions and comparison to the ground truth value of the sixth LAA of a set of six. In order: bull's eye plot representation, frontal and back views of 3D representation and bull's eye plot representation of TP, FP, TN and FN.