Medical & Biological Engineering & Computing
https://doi.org/10.1007/s11517-025-03491-y

ORIGINAL ARTICLE

Graph-Convolutional-Beta-VAE for synthetic abdominal aortic aneurysm generation

Francesco Fabbri[1] · Martino Andrea Scarpolini[1,2] · Angelo Iollo[3] · Francesco Viola[1] · Francesco Tudisco[1,4]

Received: 11 July 2025 / Accepted: 11 November 2025
© The Author(s) 2025, modified publication 2025

Abstract Synthetic data generation plays a crucial role in medical research by mitigating privacy concerns and enabling large-scale patient data analysis. This study presents a Graph Convolutional Neural Network combined with a Beta-Variational Autoencoder (GCN-β-VAE) framework for generating synthetic Abdominal Aortic Aneurysms (AAA). Using a small real-world dataset, our approach extracts key anatomical features and captures complex statistical relationships within a compact disentangled latent space. To address data limitations, low-impact data augmentation based on Procrustes analysis was employed, preserving anatomical integrity. The generation strategies, both deterministic and stochastic, manage to enhance data diversity while ensuring realism. Compared to PCA-based approaches, our model performs more robustly on unseen data by capturing complex, nonlinear anatomical variations. This enables more comprehensive clinical and statistical analyses than the original dataset alone. The resulting synthetic AAA dataset preserves patient privacy while providing a scalable foundation for medical research, device testing, and computational modeling.

Keywords Abdominal aortic aneurysms · GCN · β-VAE · Disentangled representation · Synthetic data generation

1 Introduction

In clinical practice, synthetic data plays an increasingly important role in tailoring medical treatments and therapeutic strategies to individual patients. They enable deeper understanding of disease progression and diagnostic patterns, while maintaining strict patient confidentiality and protecting sensitive health information [1–4]. Simulated or "virtual" patients are used to evaluate medical procedures and treatment strategies before they are applied in real-world clinical settings. By leveraging computational models capable of predicting clinical outcomes, researchers and clinicians can reduce uncertainty and support more informed, data-driven decisions. These tools may also enable the development of personalized treatment plans, such as custom stent-graft designs tailored to individual patient anatomies, helping to lower the risk of post-procedural complications [5–8]. Despite their potential, the validation of new interventions typically requires large and diverse patient datasets, which are often challenging to obtain due to privacy restrictions and logistical constraints. One promising solution is the generation of synthetic patient cohorts by capturing and modeling key anatomical characteristics from a limited set of real cases. These virtual populations can then be used to simulate a broader spectrum of anatomical and physiological variations, facilitating comprehensive statistical assessments and in silico clinical trials. However, generating accurate synthetic datasets from limited patient populations presents notable challenges.

Traditional linear methods, such as Principal Component Analysis (PCA), are advantageous for their simplicity and interpretability with small datasets; however, they often fail to adequately generalize beyond the limited variations captured in the original samples. On the other hand, neural network-based approaches, such as Variational Autoencoders (VAEs), are promising candidates due to their inherent nonlinear modeling capabilities [9–13]. These methods

✉ Francesco Tudisco
f.tudisco@ed.ac.uk

1 School of Mathematics, Gran Sasso Science Institute, L'Aquila 67100, Italy

2 INFN Roma Tor Vergata, Via della Ricerca Scientifica 1, Rome 00133, Italy

3 University of Bordeaux, CNRS IMB UMR 5251, Inria, centre de l'université de Bordeaux, Talence F-33405, France

4 School of Mathematics, The University of Edinburgh, Edinburgh EH9 3FD, UK

Published online: 04 December 2025

effectively capture complex anatomical variations [14–17] and potentially generalize beyond the original range of anatomical variations while still being particularly effective in representing complex features within highly compact latent spaces. As a natural evolution of the VAE framework, deep probabilistic neural networks leveraging β-VAE architectures have successfully been applied in fluid dynamics, specifically for the extraction of interpretable non-linear modes from turbulent flows [18], demonstrating superior performance compared to traditional linear methods like Proper Orthogonal Decomposition (POD) due to its ability to provide disentangled latent representations. Graph Convolutional Networks have recently been applied in diverse areas of medical research [19]. For example, Graph-Autoencoders have been employed for automatic sex determination from hip bones, significantly enhancing interpretability by explicitly encoding categorical class differences within a compact latent space [20]. Similarly, unsupervised deep learning ensemble-based approaches have been utilized for phenotyping cardiac morphology from large datasets such as the UK Biobank, improving gene discoverability related to complex heart shape variations [21].

The objective of this work is to generate realistic, synthetic abdominal aortic aneurysm (AAA) datasets that accurately capture the nonlinear anatomical variability encountered in real clinical scenarios. This is accomplished through a Graph Convolutional Network β-Variational Autoencoder (GCN-β-VAE) architecture, which leverages graph-based representations to effectively model complex anatomical shapes. By employing a novel integration of the β-VAE framework with reconstruction terms designed for mesh-based data, the latent disentanglement is enhanced, allowing targeted exploration and variation of specific morphological attributes within a compact latent space while preserving essential anatomical features.

The paper first presents a description of the dataset, followed by an in-depth explanation of the proposed GCN-β-VAE architecture, highlighting key components such as the reconstruction loss functions, spectral graph convolution, and spatial pooling sampling. A Procrustes-based data augmentation method is then introduced to enrich and diversify the training set while maintaining anatomical fidelity. The proposed methodology is evaluated through systematic benchmarking against conventional PCA-based techniques. The analysis includes the assessment of the augmentation strategy's impact, exploration of latent space properties—such as generalization capability, disentanglement, and hierarchical mode contributions—and a demonstration of the generative capabilities of the best-performing model, highlighting its potential for clinical research and in silico experimentation.

2 Methods

In Section 2.1, a description of the dataset used in this study is presented, consisting of patient-specific 3D aneurysm meshes derived from CT scans through segmentation and surface reconstruction. Next, in Section 2.2, the GCN-β-VAE Architecture (Fig. 2) is described, outlining how 3D aneurysm meshes are encoded into a latent multivariate Gaussian space and then reconstructed. In particular, the theoretical background of the VAE is addressed, along with the role of the β-hyperparameter in promoting latent disentanglement. Then, it is presented in Section 2.2.2, the reconstruction loss used to recover high-fidelity meshes, including vertex coordinate, Chamfer, normal, and edge losses (each addressing a particular geometric constraint). Subsequently, Section 2.2.3 presents the Spectral Graph Convolution approach, which leverages the eigen-decomposition of the graph Laplacian and applies Chebyshev polynomial-based filters directly on mesh vertices, while Section 2.2.4 describes the Spatial Pooling layer, highlighting how pair-contraction-based down-sampling and corresponding up-sampling matrices are used to iteratively simplify the mesh while enabling multi-resolution reconstruction. Finally, in Section 2.3, a Procrustes-Based RandAugment protocol (ProcAug) is proposed, leveraging transformations computed via Procrustes alignment, specifically scaling and rotation, to design an online augmentation procedure.

2.1 Dataset

The dataset used is composed of 60 patient-specific three-dimensional (3D) geometric reconstructions of AAAs, which were initially obtained from pre-operative CT scans of patients scheduled for abdominal aortic aneurysm treatment. These original data consist of two-dimensional gray-scale image slices from abdominal CT scans, capturing cross-sectional views of the patient's vascular anatomy [22, 23]. To construct the dataset, these 2D CT images are processed using specialized medical imaging software, *PRAEVAorta* [24], developed by NUREA, which performs automated segmentation of the vascular tree, isolating the aortic geometry from surrounding anatomical structures [25, 26]. After initial segmentation, the specific anatomical region containing the aneurysm, located below the renal arteries and extending downwards approximately 30 mm beyond the iliac bifurcation, is automatically extracted using an algorithm designed to detect and separate these anatomical landmarks [25, 26]. With the anatomical landmarks identified, the segmented aortic geometry is transformed into 3D surface meshes via the marching-cubes algorithm, which converts segmented image volumes into polygonal

surface representations [27]. The resulting surface meshes are then subjected to a smoothing procedure to remove artifacts and irregularities introduced during segmentation, utilizing the Vascular Modeling Toolkit (`vmtk`) Python library, enhancing their geometric accuracy and usability [28, 29]. By slicing it below the renal arteries and right before the iliac bifurcation, the final processed dataset thus contains detailed, patient-specific 3D surface models accurately representing the geometry of aneurysmal segments of the abdominal aorta, including features such as aneurysm sacs and associated vessel neck regions. Figure 1 visually illustrates fifteen examples of these final processed meshes, clearly depicting the anatomical variability captured by the dataset. The complete and detailed description of the entire pipeline, including image segmentation, geometry reconstruction, and surface processing steps outlined above, is described in detail in [30].

2.2 β-VAE architecture

Variational Autoencoders (VAEs) represent a class of autoencoders specifically designed to learn efficient representations or codings of input data [31]. In contrast to traditional autoencoders that produce deterministic encodings, VAEs adopt a probabilistic framework. Rather than mapping an input directly to a single point in the latent space, the encoder of a VAE maps the input to the parameters of a probability distribution p_θ parameterized by θ. A sample is then drawn from this distribution to obtain the latent representation, which is subsequently passed to the decoder. The decoder attempts to reconstruct the original input from this latent code, thereby enabling a generative modeling approach grounded in the principles of variational inference.

In practice, we aim to fit the following latent variable model to the data

$$p_\theta(x) = \int p_\theta(x|z)p_\theta(z)dz,$$

via maximum likelihood estimation (MLE) $\max \mathbb{E}_{p_{\text{data}}(x)}[\log p_\theta(x)]$. Here, $p_\theta(z)$ is the assumed prior probability distribution on the latent space, before data are taken into account, and $p_\theta(x|z)$ is the conditional probability of the data as a function of the latent space variables. Variational inference provides a variational lower bound for

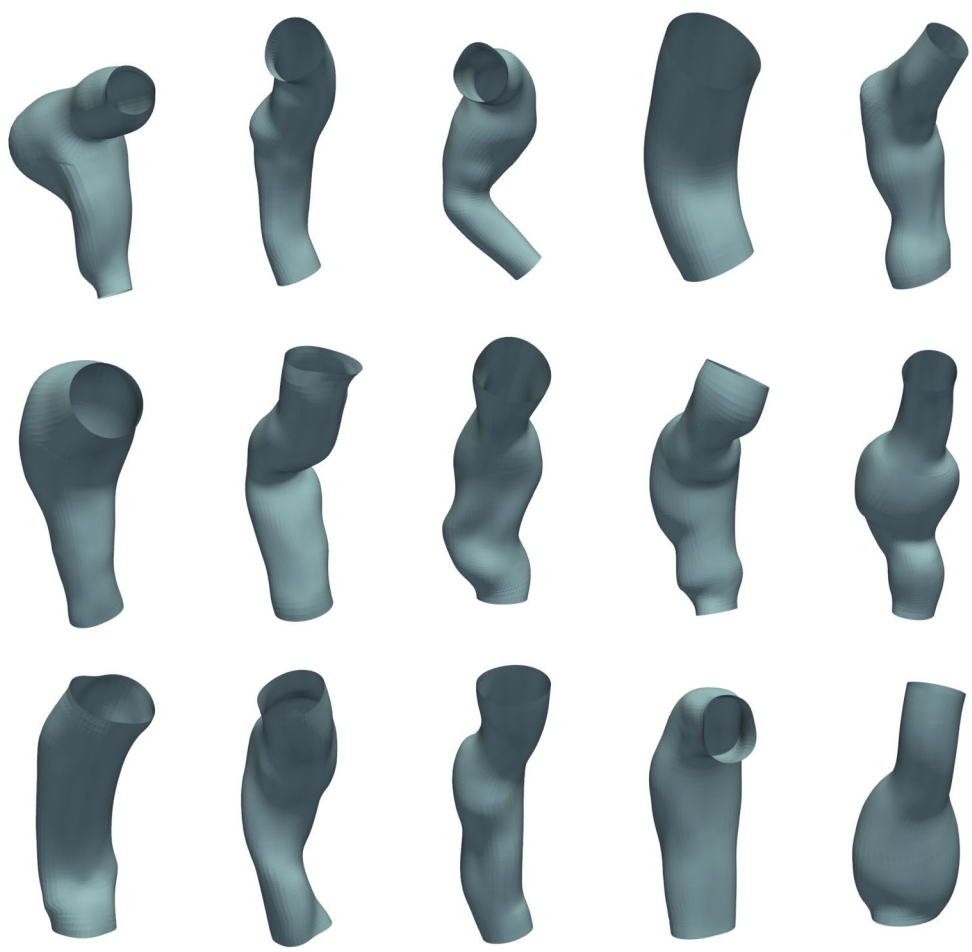

Fig. 1 Fifteen representative AAA meshes selected from the complete dataset described in Section 2.1

MLE and thus replaces MLE with the equivalent task of approximating the true posterior distribution $p_\theta(z|x)$ with a tractable distribution $q_\phi(z|x)$, parametrized by a neural network architecture forming the encoder module of the VAE (left module in Fig. 2), just like the encoder $q_\phi(z|x)$, the conditional probability $p_\theta(x|z)$ is parametrized by a neural network model forming the decoder module of the VAE architecture (right module in Fig. 2). To evaluate the statistical distance between the real and approximate posterior, the Kullback–Leibler divergence is employed,

$$D_{\mathrm{KL}}(q_\phi(z|x)||p_\theta(z|x)) = \int q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_\theta(z|x)} dz.$$

Expanding this expression using Bayes' rule gives:

$$D_{\mathrm{KL}}(q_\phi(z|x) \,\|\, p_\theta(z|x)) = -\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] + D_{\mathrm{KL}}(q_\phi(z|x) \,\|\, p_\theta(z)) + \log p_\theta(x).$$

This decomposition reveals two important terms that form the core of the variational approximation to MLE. The first term, $\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]$, is the likelihood of the observed data given the latent variables which is maximized to ensure the model accurately reproduces the input data from the latent space. The second term, $D_{\mathrm{KL}}(q_\phi(z|x) \,\|\, p_\theta(z))$, is the KL divergence between the approximate posterior $q_\phi(z|x)$ and the prior $p_\phi(z)$, which is the assumed probability distribution of the latent space. This KL term acts like a regularizer, promoting a consistent latent space by penalizing deviations from the prior. By rearranging terms, the resulting VAE loss function is expressed as:

$$\begin{aligned} L_{VAE}(\theta,\phi) &= \mathbb{E}_{p_{\mathrm{data}}(x)}\Big[D_{\mathrm{KL}}(q_\phi(z|x) \,\|\, p_\theta(z|x)) - \log p_\theta(x)\Big] \\ &= \mathbb{E}_{p_{\mathrm{data}}(x)}\Big[-\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] \\ &\qquad + D_{\mathrm{KL}}\big(q_\phi(z|x) \,\|\, p_\theta(z)\big)\Big]. \end{aligned} \tag{1}$$

In the context of Variational Bayesian methods, the VAE loss corresponds to what is known as the evidence lower bound (ELBO). The KL divergence is always non-negative, and therefore $-L_{VAE}(\theta,\phi)$ is the lower bound of $\log p_\theta(x)$:

$$\begin{aligned} -L_{VAE}(\theta,\phi) &= \mathbb{E}_{p_{\mathrm{data}}(x)}\left[\log p_\theta(x) - D_{\mathrm{KL}}(q_\phi(z|x) \,\|\, p_\theta(z|x))\right] \\ &\le \mathbb{E}_{p_{\mathrm{data}}(x)}\left[\log p_\theta(x)\right]. \end{aligned}$$

Therefore, minimizing the loss function means maximizing the lower bound probability of generating realistic data.

2.2.1 Sampling in the latent space

The expectation term in the loss function invokes generating samples from $z \sim q_\phi(z|x)$. Sampling is inherently a stochastic process, which prevents straightforward gradient backpropagation. To enable end-to-end training, the reparameterization trick is employed. The encoder $q_\phi(z|x)$ is modeled to output a pair of parameters $q_\phi : x \longrightarrow (\mu, \sigma)$ for a given input x, representing the mean and standard deviation of a multivariate Gaussian distribution. The approximate posterior is defined as $z \sim q_\phi(z|x) = \mathcal{N}(\mu, \sigma^2 I)$,

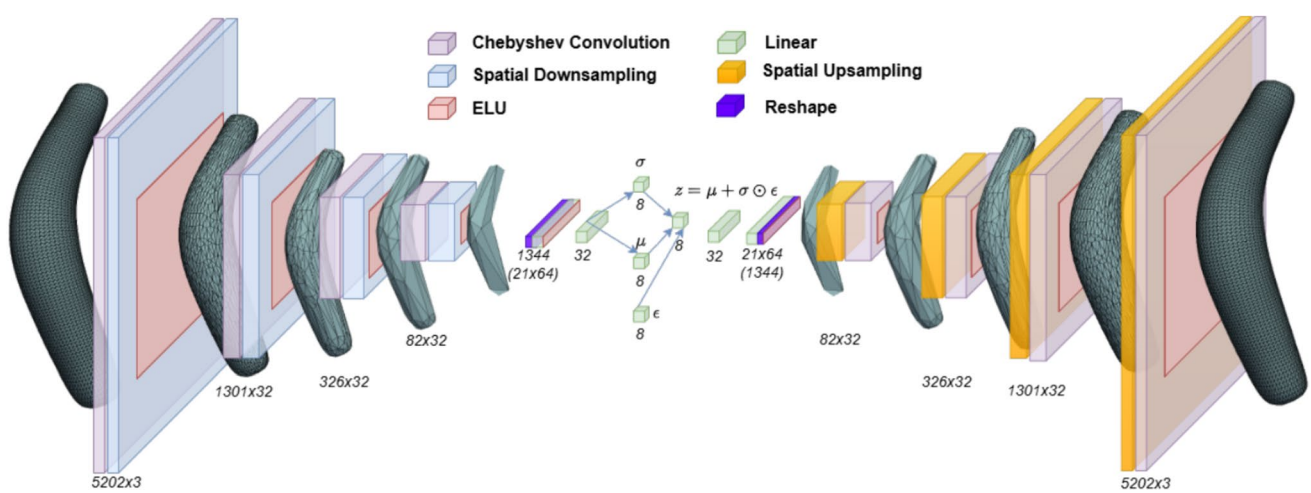

Fig. 2 Architecture of GCN-β-VAE used in the generation experiments. Meshes are downsampled and upsampled by a factor of 4 in each Spatial pooling layer (Section 2.2.4). Spectral graph convolutions, implemented using Chebyshev polynomials (Section 2.2.3), are applied to the mesh data, transforming the 3 input channels through successive convolutional layers with output feature dimensions of $32 \to 32 \to 32 \to 64$. Additionally, linear layers are applied in the hidden layers preceding and following the latent space, as well as within the latent space itself. An Exponential Linear Unit (ELU) serves as the activation function throughout the network

while the prior distribution is set to a standard multivariate Gaussian $z \sim p_\theta(z) = \mathcal{N}(0, I)$.

The reparameterization trick enables sampling by expressing z as:

$$z = \mu + \sigma \odot \epsilon, \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, I), \tag{2}$$

where $\odot$ denotes element-wise multiplication. This formulation allows the model to learn the parameters μ and σ through backpropagation, while maintaining stochasticity via the auxiliary noise variable ϵ. The reparameterization trick is implemented by the green linear layers in Fig. 2, which operate within the latent space of the VAE. By Substituting the Gaussian prior $p_\theta(z) = \mathcal{N}(0, I)$ and the approximate posterior $q_\phi(z \mid x) = \mathcal{N}(\mu, \sigma^2 I)$ into the KL divergence term yields the final expression for the VAE loss.

$$L_{\text{VAE}}(\theta, \phi) = -\frac{1}{2} \sum_{i=1}^{d} \left(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2\right) - \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log p_\theta(x|z)\right].$$

The proposed β-VAE [32] is a modification of the Variational Autoencoder discussed above, with a special emphasis on discovering disentangled latent factors. Following the same arguments as for the VAE, the objective is to maximize the probability of generating real data while keeping the distance between the real and estimated posterior distributions small. To achieve this, an additional hyperparameter β is introduced as a weighting factor on the KL divergence term. Proper tuning of β favors a more orthogonal and disentangled latent space representation:

$$L_{\text{VAE}}(\theta, \phi) = -\frac{\beta}{2} \sum_{i=1}^{d} \left(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2\right) - \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log p_\theta(x|z)\right]. \tag{3}$$

Increasing the value of β encourages the model to learn such efficient and disentangled latent representations. However, if β is set too high, it may introduce a trade-off, where improved disentanglement comes at the cost of degraded reconstruction quality.

2.2.2 Combining regularization and reconstruction loss

The chosen training loss combines the regularization KL-based term from the β-VAE, as described above, with a reconstruction component that encourages the generated outputs to match the geometries present in the training data. The reconstruction component $\mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z)$ is formulated as a weighted combination of multiple loss terms, each capturing different geometric aspects of the 3D mesh structure. This composite design allows for flexibility in adapting the loss to the specific characteristics of the data and the reconstruction task. By exploiting the inherent connectivity of mesh graphs, higher-order loss terms are defined over local neighborhoods of vertices, enabling more effective regularization of complex 3D shapes. A combination of vertex, Chamfer, normal, and edge terms was chosen, motivated by their complementary roles in mesh reconstruction. Vertex and Chamfer distance provide global correspondence measures well suited for shape analysis, whereas normal consistency and edge-length penalties act as lightweight yet effective local regularizers, promoting smoothness and orientation coherence [33, 34]. Alternatives, such as Laplacian regularization, were also evaluated. These did not yield clear improvements relative to their higher computational cost. Formally, the reconstruction terms are defined as follows:

- Vertex Coordinate Loss: Minimizes the L1 distance between the predicted 3D mesh coordinates M and the ground truth coordinates M^*, providing a robust constraint that penalizes local deviations in vertex positions:

$$L_{\text{vertex}} = \|M - M^*\|_1,$$

where M^* denotes the ground truth mesh coordinates, $\|\cdot\|_1$ represents the L1 norm.

- Chamfer Loss: Computes how far each point in one set is from the closest point in the other set, which ensures that the reconstructed shape closely matches the overall structure and distribution of the target geometry:

$$L_{\text{chamfer}} = \sum_{x \in M} \min_{y \in M^*} \|x - y\|_2^2 + \sum_{y \in M^*} \min_{x \in M} \|y - x\|_2^2,$$

where the term $\|\cdot\|_2$ denotes the L2 norm, used to calculate the Euclidean distance between two vertices.

- Normal Loss: Enforces the consistency between the normal vectors of the predicted mesh and the corresponding ground truth normals, thereby promoting smooth surfaces and enhanced local detail. The surface normal loss, L_{normal}, is defined as follows:

$$L_{\text{normal}} = \sum_{f} \sum_{\{i,j\} \subset f} \left\langle \frac{x_i - x_j}{\|x_i - x_j\|_2}, n_f^* \right\rangle,$$

where f represents a triangle face on the mesh, and n_f^* is the unit normal vector of the corresponding ground truth

face. The terms x_i and x_j refer to the i-th and j-th vertices of the triangle face f. This loss enforces the edge between a vertex and its neighbours to be perpendicular to the surface normal from the ground truth.

- Edge Loss: Enforces the consistency of edge lengths between the predicted mesh and the ground truth mesh. This loss is particularly effective for recovering smoothness in regions with dense vertices, as well as penalizing flying vertices, which can cause long edges. The edge loss L_{edge} is defined as follows:

$$L_{\text{edge}} = \sum_{f} \sum_{\{i,j\} \subset f} \left| \|x_i - x_j\|_2 - \|x_i^* - x_j^*\|_2 \right|,$$

where f represents a triangle face on the mesh, x_i and x_j are the vertices of the predicted mesh, and x_i^* and x_j^* denote the corresponding vertices of the ground truth mesh.
The final loss is then written as:

$$\begin{aligned} L(\theta, \phi) = &-\frac{\beta}{2} \sum_{i=1}^{d} \left(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2\right) \\ &+ \alpha_{\text{vertex}} L_{\text{vertex}} + \alpha_{\text{chamfer}} L_{\text{chamfer}} \\ &+ \alpha_{\text{edge}} L_{\text{edge}} + \alpha_{\text{normal}} L_{\text{normal}}. \end{aligned} \tag{4}$$

Here, the coefficients $\{\alpha_{\text{label}}\}$ are hyperparameters that regulate the trade-off between the contributions of the individual loss terms, determining the balance between global correspondence measures (vertex, Chamfer) and local regularization terms (edge, normal). During training, these are set as $\alpha_{\text{chamfer}} = \alpha_{\text{vertex}} = 1$ and $\alpha_{\text{edge}} = \alpha_{\text{normal}} = 0.1$ for all configurations of latent size and β. The coefficients were initialized following prior literature, Wang et al. [33] and Choi et al. [34], and then tuned on our dataset to optimize reconstruction accuracy. The final selection provided consistent and accurate reconstructions across all k-fold splits.

The entire β-VAE architecture and loss formulation were implemented on top of open-source libraries. In particular, `PyTorch` and `PyTorch3D` were used for deep learning and geometric operations, while `Torch Geometric` was employed for the graph-based interpretation of meshes. Additionally, `MPI-IS Mesh` processing tools were used for mesh handling.

2.2.3 Spectral graph convolution

A 3D aneurysm mesh is represented as an undirected graph $M = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{v_i\}_{i=1}^{N}$ is the set of $N = |\mathcal{V}|$ edges vertices, $\mathcal{E} = \{e_j\}_{j=1}^{E}$ the set of $E = |\mathcal{E}|$ edges, and $\mathcal{A} = (\mathrm{a}_{ij})_{N \times N}$ the adjacency matrix, where $\mathrm{a}_{ij} = 0$ if $(i,j) \notin \mathcal{E}$ and $\mathrm{a}_{ij} \neq 0$ if $(i,j) \in \mathcal{E}$. One can also compute the symmetric normalized graph Laplacian as $\mathcal{L} = \mathcal{I}_N - (\mathcal{D}^+)^{1/2} \mathcal{A} (\mathcal{D}^+)^{1/2}$ where $\mathcal{D}^+$ is the Moore-Penrose inverse of the diagonal matrix $\mathcal{D} = diag(\sum_j \mathrm{a}_{ij})$ and $\mathcal{I}_N$ is the identity matrix of size N. The normalized Laplacian in its spectral basis has the form $\mathcal{L} = \mathcal{U} \Lambda \mathcal{U}^T$ where $\mathcal{U} \in \mathbb{R}^{N \times N}$ is the matrix of the orthonormal eigenvectors of $\mathcal{L}$ and Λ is the matrix of the eigenvalues of $\mathcal{L}$. A signal on the nodes (vertices) of the mesh graph is defined as $\mathcal{S} \in \mathbb{R}^{N \times F}$ where the number of features F, before applying any of the convolutional filter operations, is equal to 3, which is equal to the number of dimensions in which the mesh exists. The value $\mathcal{S}^j$ is the value of the jth feature element of $\mathcal{S}$ on the nodes. The fast spectral filtering method [35], previously used in Ranjan et al. [19], is employed. An input feature on a node $\mathcal{S}_{F_{in}}$ is transformed into $\mathcal{S}_{F_{out}}$ using a recursive filter kernel f based on Chebyshev polynomials of order K, as follows:

$$f_\theta(\mathcal{L}) = \sum_{k=0}^{K-1} \theta_k \mathcal{T}_k(\bar{\mathcal{L}}),$$

where $\bar{\mathcal{L}}$ is the shifted and rescaled Laplacian $\bar{\mathcal{L}} = 2\mathcal{L}/\lambda_{max} - I_N$, normalized by its largest eigenvalue λ_{max} and shifted by the identity matrix; $\theta \in \mathbb{R}^K = (\theta_0, \ldots, \theta_{K-1})$ is a vector of learnable coefficients; and $\mathcal{T}_k$ denotes the k-th Chebyshev polynomial of the first kind, defined recursively as $\mathcal{T}_k(x) = 2x\mathcal{T}_{k-1}(x) - \mathcal{T}_{k-2}(x)$ with $\mathcal{T}_1 = x$ and $\mathcal{T}_0 = 1$. The output signal would be written as:

$$\mathcal{S}_{F_{out}}^j = \sum_{i=1}^{F_{in}} f_{\theta_{i,j}}(\mathcal{L}) S_{F_{in}}^i,$$

where F_{in} denotes the number of input features per node. The index j runs over the output features being computed by the filter, the j-th output feature channel $S_{F_{\text{out}}}^j$ is formed by summing over all F_{in} input channels. Each term in the sum consists of the Chebyshev-based filter $f_{\theta_{i,j}}(\mathcal{L})$, which transforms the i-th input channel to the j-th output channel, multiplied by the i-th input feature $S_{F_{\text{in}}}^i$. This spectral convolution operation, based on Chebyshev polynomials, is implemented as the purple Chebyshev convolution layers in the architecture diagram Fig. 2.

2.2.4 Spatial pooling

Spatial pooling is a down-sampling operation commonly used to combine small-scale (local) and large-scale (global) information by reducing the spatial resolution of data, typically in image-based neural networks. In essence, pooling

summarizes local information to create progressively simplified representations. However, for mesh data, spatial pooling becomes more complex because meshes do not have a regular grid structure like images, and thus, simplification must respect geometric and topological constraints. Spatial pooling on meshes is implemented using a simplification algorithm based on iterative vertex pair contraction [36]. Specifically, a pair contraction, represented as $(v_1, v_2) \rightarrow \bar{v}$, merges two vertices v_1 and v_2 to a new position $\bar{v}$, connects all their incident edges to v_1, and removes the vertex v_2. Subsequently, any edges or faces that become degenerate are removed. Because contractions are highly localized operations, applying them iteratively generates a series of simplified models $M_N, M_{N-1}, \ldots, M_g$, starting from the initial model M_N down to a final approximation M_f.

The in-network down-sampling method from Ranjan et al. [19] is implemented on a mesh initially containing N vertices, using transformation matrices $Q_d \in 0, 1^{n \times N}$ for down-sampling and $Q_u \in \mathbb{R}^{N \times n}$ for up-sampling, where $N > n$. In the architecture diagram Fig. 2, the matrix-based down-sampling and up-sampling operations performed via Q_d and Q_u, respectively are implemented as the azure-colored downsampling layers and the yellow upsampling layers, which appear symmetrically on the encoder-decoder structure. The down-sampling process iteratively merges vertex pairs while preserving surface detail through quadric error matrices. After down-sampling, the resulting vertex set V_d forms a subset of the original set V, such that $V_d \subset V$. The elements of $Q_d(p, q)$ indicate whether the q-th vertex is retained ($Q_d(p, q) = 1$) or discarded ($Q_d(p, q) = 0$ for all p). Since perfect down-sampling and up-sampling cannot be achieved for arbitrary surfaces, the up-sampling matrix is designed concurrently with the down-sampling step. The vertices retained during down-sampling undergo transformations and are similarly retained during up-sampling, where $Q_u(q, p) = 1$ if $Q_d(p, q) = 1$. Vertices that are discarded during the down-sampling stage are mapped back onto the simplified mesh surface using barycentric coordinates. The discarded vertex v_q is projected onto the nearest triangle formed by vertices in V_d, and its position is calculated using the barycentric formula $v_{\text{ep}} = w_i v_i + w_j v_j + w_k v_k$, where $v_i, v_j, v_k \in V_d$ and the weights satisfy $w_i + w_j + w_k = 1$. These weights are then used to update the up-sampling matrix as $Q_u(q, i) = w_i$, $Q_u(q, j) = w_j$, and $Q_u(q, k) = w_k$, while setting $Q_u(q, l) = 0$ for all other vertices. The final up-sampled mesh V_u is obtained through sparse matrix multiplication, $V_u = Q_u V_d$.

2.3 ProcAug augmentation

In many computer vision tasks, data augmentation has proven to be an essential technique for improving model robustness and generalization. Augmentation pipelines rely on elementary transformations, such as random rotations, random crops, flips, and jittering. Recent works have moved toward more sophisticated and automated augmentation strategies, for example, AutoAugment [37] and RandAugment [38], where a search algorithm or a simplified policy is employed to discover optimal augmentations.

Procrustes analysis [39, 40] has gained attention in shape analysis due to its ability to align and compare data by reducing the effects of translations, scales, and rotations. More specifically, it seeks a transformation that best superimposes one set of points onto another, effectively capturing the intrinsic shape while normalizing nuisance factors. This property makes Procrustes analysis a powerful tool for quantifying and transferring geometric transformations across datasets, as illustrated in Fig. 3, this alignment significantly reduces variability in shape representations by minimizing reconstruction errors such as Chamfer Distance and L2 error distributions..

Procrustes analysis and data augmentation can be understood as two sides of the same coin in that they deal with the same transformations, translations, rotations, and scalings, but in opposite ways. Procrustes analysis identifies and removes these variations to align data points in a common coordinate system, effectively standardizing the shapes so that their intrinsic structure is more directly comparable. While this process reduces the variability of the dataset, it can also remove some precious morphological details that may be essential for capturing the true range of anatomical differences in AAA. Data augmentation, on the other hand, deliberately applies these transformations (and others, such as jitter or translations) to existing data points, thereby increasing variability in a controlled way. The aim of augmentation is not to eliminate transformations but to introduce them, expanding the training set and making models more robust to real-world variations. Thus, while Procrustes analysis seeks to unify or harmonize data by subtracting differences, data augmentation enlarges the feature space by creating new instances that share essential properties of the original data, ultimately illustrating how each approach works with the same underlying transformations but pursues fundamentally opposite goals.

At a high level, the idea is to extract real-valued transformations (e.g., rotation angles, scaling factors) from data alignment. These transformations can then be randomly sampled and applied to incoming batches during training. This approach blends the flexibility of RandAugment, where specific augmentation types can be randomly selected, with the geometrically grounded transformations from Procrustes analysis. The result is a method that can preserve the underlying structure of meshes while still introducing valuable variability that encourages the model to learn more robust representations.

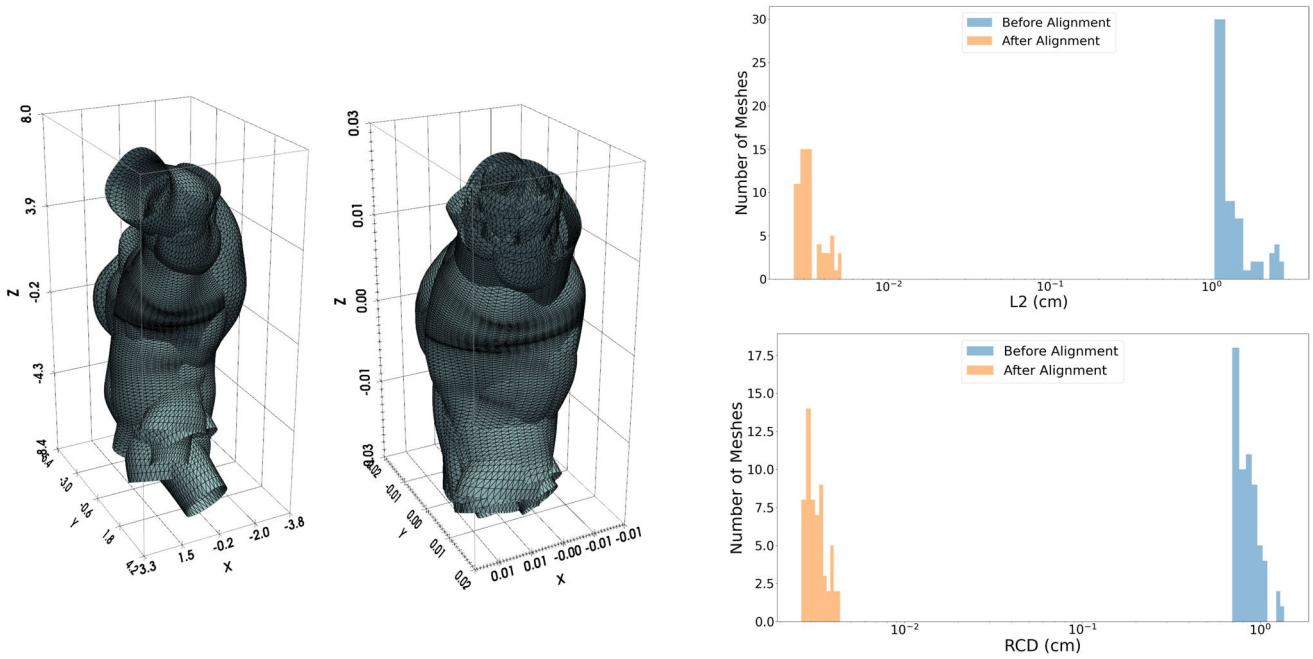

Fig. 3 Original real meshes (left) Corresponding meshes after Procrustes alignment (right). The alignment removes differences in scale, rotation, and translation, bringing the meshes into a common reference frame. **(b)** Histograms comparison of reconstruction errors before and after Procrustes alignment. The plots show the distribution of Chamfer Distance (bottom) and L2 error (top) for meshes before alignment and after alignment. After Procrustes alignment, both error distributions become more concentrated around zero, indicating a reduction in variability

Procrustes analysis normalize the AAA mesh, represented by the matrix $M \in \mathbb{R}^{N\times 3}$ to have unit norm $M = \frac{M}{\|M\|_F}$ as well as the reference geometry $M_{ref} = \frac{M_{ref}}{\|M_{ref}\|_F}$ · Then, Procrustes analysis solves the orthogonal Procrustes problem by minimizing:

$$\min_{\Omega} \|M\Omega - M_{ref}\|_F, \tag{5}$$

where Ω is the rotation matrix which most closely maps M to M_{ref}. This problem admits an explicit solution [39]. Using the Singular Value Decomposition (SVD), matrices U, Σ, and V^T are obtained as follows: $U, \Sigma, V^T = \mathrm{SVD}(M^T M_{ref})$, where U and V are orthonormal and Σ is diagonal. Thus the optimal rotation matrix Ω^* is given by:

$$\Omega^* = UV^T\,. \tag{6}$$

Both the information on the scaling, as well as the rotation can be used as data augmentation parameters, in fact in 3D, a rotation can be parameterized by three Euler angles [41] (e.g., ψ, ξ, γ) around the x, y, and z axes, respectively. Suppose we observe or estimate that each shape M_i is rotated by angles within some range w.r.t. the reference shape:

$$\psi_{\min} \le \psi_i \le \psi_{\max},$$
$$\xi_{\min} \le \xi_i \le \xi_{\max},$$
$$\gamma_{\min} \le \gamma_i \le \gamma_{\max}.$$

Each value of Euler angles can then be sampled and utilized to rotate the training geometry.

$$\psi_{\text{rand}} \sim \mathcal{U}(\psi_{\min}, \psi_{\max}),$$
$$\xi_{\text{rand}} \sim \mathcal{U}(\xi_{\min}, \xi_{\max}),$$
$$\gamma_{\text{rand}} \sim \mathcal{U}(\gamma_{\min}, \gamma_{\max}).$$

where $\mathcal{U}$ denotes a uniform distribution. The resulting 3D rotation matrix, Ω_{rand}, is constructed via the standard Euler angle to rotation matrix composition. For instance, selecting the the ZYX Euler angles ordering:

$$\Omega_{\text{rand}} \;=\; \Omega_z(\gamma_{rand})\,\Omega_y(\xi_{rand})\,\Omega_x(\psi_{rand}),$$

where each $\Omega_x(\psi), \Omega_y(\xi), \Omega_z(\gamma)$ is a 3×3 rotation matrix about its respective axis. Finally, the augmented norm is defined:

$$M_{aug} = M_i \Omega_{\text{rand}}^T.$$

Similarly, a range of values is defined based on the norms:

$$s_{\min} = \|M\|_{\min} \leq \|M_i\| \leq \|M\|_{\max} = s_{\max},$$

where $\|M\|_{\max}$ and $\|M\|_{\min}$ are the largest and smallest norm values among the training set, and sample them during training

$$s_{\text{rand}} \sim \mathcal{U}(s_{\min}, s_{\max}),$$

the augmented data is then written as:

$$M_{aug} = M_i \frac{s_{rand}}{\|M_i\|}.$$

This approach allows to systematically generate augmented training samples by varying both scale and orientation, enhancing the robustness of the model. Moreover, inspired by RandAugment [37], a parameter-free augmentation procedure is adopted to simplify policy search while maintaining diversity in the augmented samples. Instead of learning augmentation policies and probabilities for each transformation, each transformation is applied with a uniform probability of $\frac{1}{K}$, where $K = 3$, including identity, scaling, and rotation.

3 Results and discussion

The following section evaluates the performance of the model and outlines the construction of a meaningful latent space suitable for generative experiments. The analysis begins with the impact of the augmentation strategy, ProcAug, in Section. 3.1. From Sections 3.2 to 3.2.2, key properties of the latent space are systematically investigated. Specifically, Section 3.2 assesses the model's out-of-sample generalization capability by comparison with a baseline PCA approach. Section 3.2.1 examines the disentanglement properties of the learned latent space, focusing on the statistical independence of its representations. Section 3.2.2 introduces the concept of hierarchical latent mode contribution to identify how different latent dimensions capture distinct modes of variation in the data. Finally, based on insights from these analyses, the best-performing model identified in earlier sections is used for generative in-sample experiments in Section 3.3.

Given the frequent shortage of data encountered in the medical field, the ability of machine learning models to generalize effectively from limited datasets is critical. Before evaluating the generative capacity of the model, its generalization capabilities are first discussed, as these capabilities determine the accuracy on unseen data and potential applicability in real-world scenarios. Accordingly, k-fold validation is conducted on all metrics presented in the following section with $k = 10$, averaging the values obtained from each split. This out-of-sample analysis importantly guides the subsequent generation of novel and meaningful data.

Specifically, the trained GCN-β-VAE model is utilized on the in-sample dataset to produce new data. The ultimate goal is to derive a latent representation of AAA that is compact, disentangled, and capable of accurately reconstructing unseen data, thereby validating the effectiveness of the model's generative and representational capacities. During reconstruction, the reparameterization trick (Eq. 2) simplifies to using only the mean, $\boldsymbol{z} = \boldsymbol{\mu}$.

The quality of reconstruction serves as an essential measure of the model's ability to capture the underlying geometry and topology of the input data. To quantify reconstruction quality, both the L_2 norm and the L_2 reconstruction percentage are employed, defined as follows:

$$E = \left(1 - \frac{\|M - M^*\|_2}{\|M^*\|_2}\right) \times 100 \quad \text{where } \|x\|_2 = \sqrt{\sum_i x_i^2}, \tag{7}$$

where M^* represents the set of original data points and M represents the set of reconstructed data points. Additionally, the Chamfer distance [42, 43], already defined in the loss function Eq. 4, is employed and treated as an absolute error:

$$\text{Err}_{\text{CD}} = \sum_{x \in M} \min_{y \in M^*} \|x - y\|_2^2 + \sum_{y \in M^*} \min_{x \in M} \|y - x\|_2^2. \tag{8}$$

To enhance interpretability and comparability across metrics, the Root Chamfer Distance (RCD) is also reported, defined as:

$$RCD = \sqrt{\text{Err}_{\text{CD}}}, \tag{9}$$

thus making it more directly comparable to the L_2 norm.

To evaluate the stastical independence of the latent representation variables, the KL-Divergence defined previously in Eq. 4 is employed, as well as the determinant of the correlation matrix $\mathrm{R} \in \mathbb{R}^{d \times d}$ times 100, denoted as $\det(\mathrm{R})$, which is defined as:

$$\mathrm{R} = (R_{\text{ij}})_{d \times d}, \tag{10}$$

with components:

$$R_{ij} = \begin{cases} 1 & \text{if } i = j, \\ \frac{C_{ij}}{\sqrt{C_{ii} C_{jj}}} & \text{if } 1 \leq i \neq j \leq d, \end{cases}$$

where d is the latent size dimension and C_{ij} represents the (i, j)-th component of the covariance matrix $\mathrm{C} \in \mathbb{R}^{d \times d}$ defined as:

$$\mathrm{C} = \mathbb{E}\left[(\mathrm{X} - \mathbb{E}[\mathrm{X}])(\mathrm{X} - \mathbb{E}[\mathrm{X}])^T\right].$$

Where, given n observations of the latent space $\{\boldsymbol{\mu}^{(i)}\}_{i=1}^{n} \subset \mathbb{R}^d$, one can define, the latent matrix: $\mathrm{X} \in \mathbb{R}^{n \times d}$ where each row is an observation and each column is a variable of the latent space.

3.1 ProcAug performance

The impact of ProcAug is evaluated in an ablation study, in which individual components such as scaling and rotation are selectively removed to assess their contribution, using two metrics: Euclidean Distance (L_2) and RCD (Eq. 9). Table 1 summarizes these results on the test dataset, averaged over 10-fold cross-validation. The results indicate that ProcAug consistently improves reconstruction accuracy, achieving the lowest errors across all latent sizes. While applying only scaling or only rotation individually also leads to improvements over the baseline (No ProcAug), combining both transformations within ProcAug yields the most significant performance gains. Specifically, ProcAug achieves a relative reduction in L_2 norm, ranging from 8.46% to 18.90% and in RCD from 6.70% to 15.78%, compared to the baseline. This demonstrates that the synergistic effect of jointly applying rotation and scaling enhances the robustness and accuracy of the learned representations more than either transformation alone.

Furthermore, Fig. 4 illustrates the evolution of a test reconstruction term (k-folded 10 times) in the loss function Eq. 4 over the training epochs for different latent sizes,

Table 1 Comparison of the reconstruction performance of VAE (Euclidean norm and RCD) trained with different data-augmentation procedures

L_2 (cm)					
Latent size	ProcAug	Only scaling	Only rotation	No ProcAug	% Diff (ProcAug)
4	0.376	0.447	0.436	0.447	**-15.93%**
8	0.251	0.258	0.272	0.275	**-8.46%**
12	0.224	0.226	0.241	0.254	**-11.96%**
16	0.207	0.209	0.221	0.237	**-12.69%**
20	0.195	0.197	0.212	0.222	**-12.22%**
24	0.184	0.194	0.196	0.227	**-18.90%**
RCD (cm)					
4	0.373	0.427	0.421	0.431	**-13.52%**
8	0.258	0.267	0.285	0.277	**-6.70%**
12	0.237	0.240	0.248	0.260	**-8.99%**
16	0.224	0.234	0.234	0.254	**-11.88%**
20	0.218	0.222	0.229	0.245	**-10.90%**
24	0.209	0.220	0.220	0.248	**-15.78%**

The results show the mean over 10 random foldings

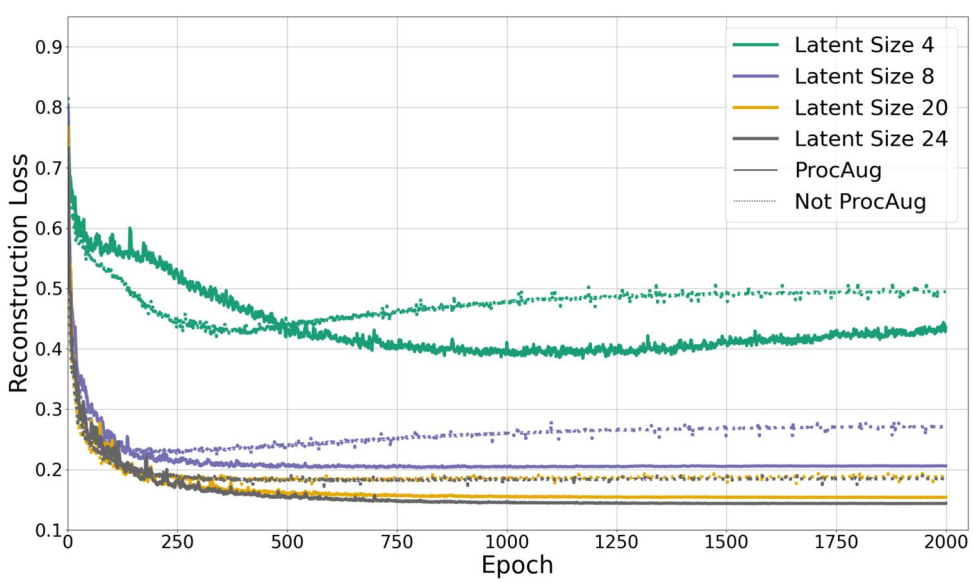

Fig. 4 Test Reconstruction error vs. epochs for different latent sizes, comparing ProcAug and NoProcAug. ProcAug consistently achieves lower reconstruction error and convergence across various latent sizes. Moreover, NoProcAug exhibits spikes in the error curve; in contrast, ProcAug, especially for higher latent sizes, results in smoother curves, indicating a more stable reconstruction performance. The results are k-folded 10 times

comparing ProcAug and NoProcAug. The figure reinforces the findings from Table 1, showing that ProcAug not only achieves lower final reconstruction error but also accelerates convergence across different latent sizes.

3.2 Out-of-sample reconstruction

Similarly to PCA, autoencoder-based methods project data onto a lower-dimensional latent space and then provide the inverse transformation. While the first finds eigenvectors of the covariance matrix to maximize variance in the projected data, the latter uses a nonlinear neural network to learn a latent representation by minimizing a loss function.

Moreover, PCA learns an orthogonal basis using training data, projecting the test set onto it and then uses the transpose principal components matrix to map back the reduced representation to the original feature space. On the other hand, autoencoder-based model maps the test set onto the feature space using a decoder (not necessarily symmetric to the encoder) block that performs upsampling. To accurately evaluate parallelism, PCA is compared against different β -VAE models, each trained with varying latent dimension sizes ranging from $d = 4, 8, \ldots, 24$. The reconstruction results, considering $\beta = 1.0e-3$, are depicted in Fig. 5. While the pointwise L_2 reconstruction error is comparable to PCA, this metric alone does not fully represent reconstructed shape quality. Minimizing L_2 does not necessarily guarantee good shape preservation, particularly for anatomical structures. In fact, two shapes with similar pointwise distances can still significantly differ in overall geometry and topology [44, 45]. The GCN-β-VAE notably outperforms PCA regarding Chamfer error, a geometry-aware metric, indicating superior shape preservation. This distinction is crucial, as anatomical and functional interpretations rely significantly on accurate geometry.

3.2.1 Latent analysis and disentangled representation

Unsupervised β-VAE relies on identifying independent latent representations through the influence of the Lagrangian multiplier β on the KL-divergence constraint defined in Eq. 3. By forcing the estimated posterior distribution $q_\phi(z|x)$ to resemble the prior distribution $p_\theta(z) \sim \mathcal{N}(0, I)$, the latent representation is encouraged to approximate a standard multivariate Gaussian characterized by zero mean and identity covariance. This behavior is demonstrated in Fig. 6, where distributions of latent variables progressively approach a standard Gaussian, attaining unit variance and reduced correlation with other variables as β increases.

This trend indicates that the regularization induced by β impacts the compactness of the latent space. Specifically, given a fixed latent dimension, only a subset of latent variables is necessary to adequately represent the entire dataset. This concept, termed inner latent compactness, is further

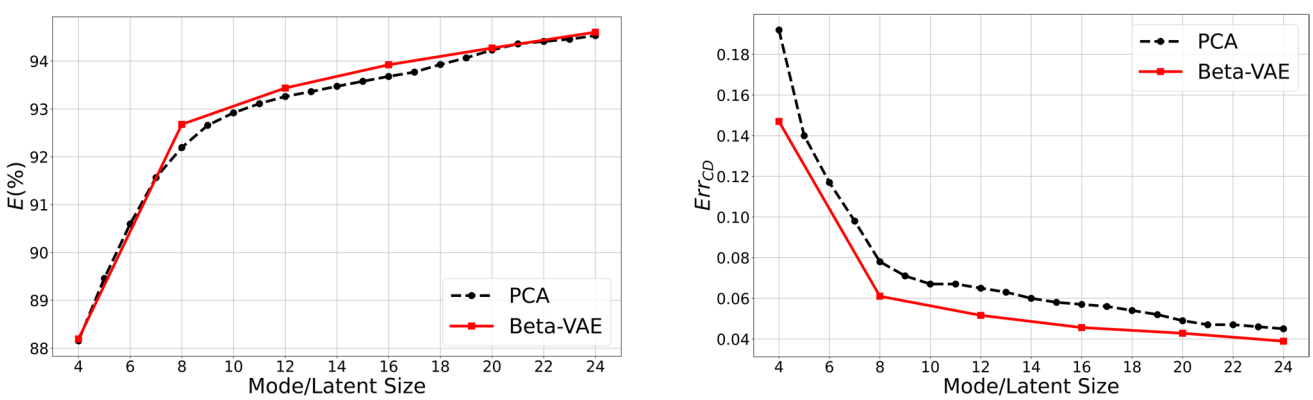

Fig. 5 Behavior of our trained model with different latent size dimensions vs PCA for L_2 reconstruction error Eq. 7 (a) and Chamfer error Eq. 8 (b)

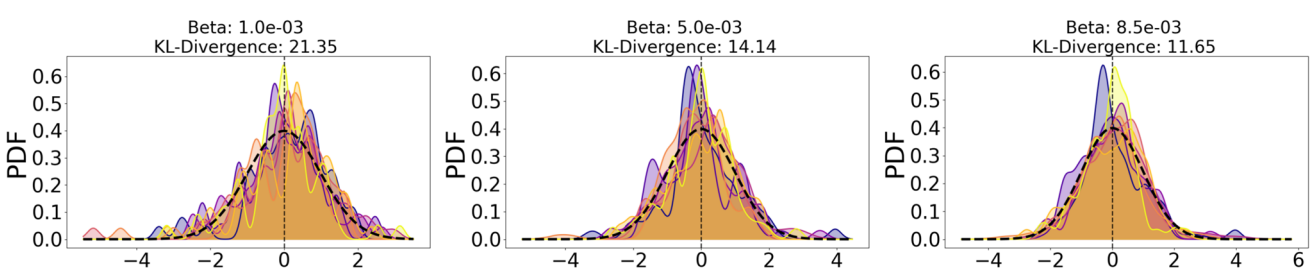

Fig. 6 Probability density functions for each variable in the latent space (colored continuous curves) and the standard Gaussian (dotted curve). As β increases (left to right), the KL-divergence between the estimated posterior and the prior decreases

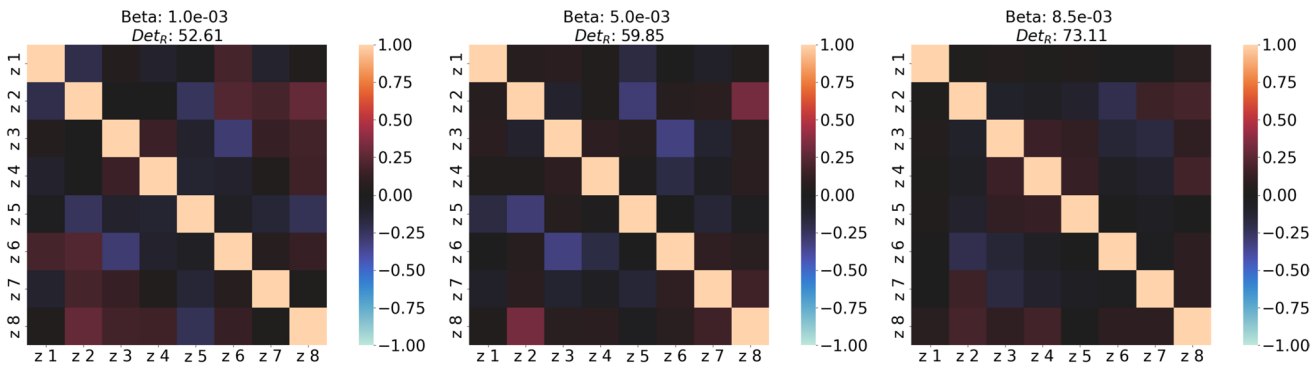

Fig. 7 Correlation matrices of the latent variables for different values of beta. The colorbar indicates the value of each element of the matrix

explored in Section 3.2.2. To evaluate the independence among latent variables across different β values, the correlation matrix of these variables is analyzed, as illustrated in Fig. 7, which reports the determinant values across varying β. Increasing β results in a higher determinant of the correlation matrix, pushing it closer to the identity matrix. This effect, recognized for promoting more interpretable latent variables [46, 47], slightly compromises reconstruction accuracy.

Reconstruction accuracy and correlation matrix determinants across varying β values and latent dimensions are presented in Fig. 8, which shows the trade-off between reconstruction error and latent disentanglement. When increasing β from $1.0e-03$ to $8.5e-03$, the determinant of the correlation matrix improves from $\det(\mathrm{R}) = 52.61$ to $\det(\mathrm{R}) = 73.11$, while reconstruction percentage marginally declines from $E = 91.90\%$ to $E = 91.48\%$. Nevertheless, this performance consistently surpasses PCA, which achieves a reconstruction percentage of $E = 91.19\%$.

The behavior of the reconstruction percentage and $\det(\mathrm{R})$ is also shown as a function of the latent size. As the latent size increases, reconstruction accuracy improves significantly, ranging from $E = 88.19\%$ at latent size $d = 4$ to $E = 94.61\%$ at $d = 24$. This outcome aligns with expectations, as a larger latent space enables the network to encode more information, reducing loss during the encoding process. However, the increased capacity also introduces the risk of encoding semantically irrelevant details or "noise" rather than focusing on independent statistical factors. This effect is mirrored in the determinant $\det(\mathrm{R})$, which decreases from 89.33 at $d = 4$ to 0.92 at $d = 16$. The decline in determinant values for higher-dimensional latent spaces reflects the model's greater freedom to encode detailed input characteristics, potentially resulting in correlated noise or redundancies instead of purely salient, independent features.

3.2.2 Hierarchical inner latent contribution

In Section 3.2.1, it is shown that both PCA and a VAE with increasing latent dimensionality can capture similar amounts of variance. Unlike PCA, which naturally ranks its principal components by explained variance, a VAE does not impose an inherent order on its latent dimensions. This distinction is

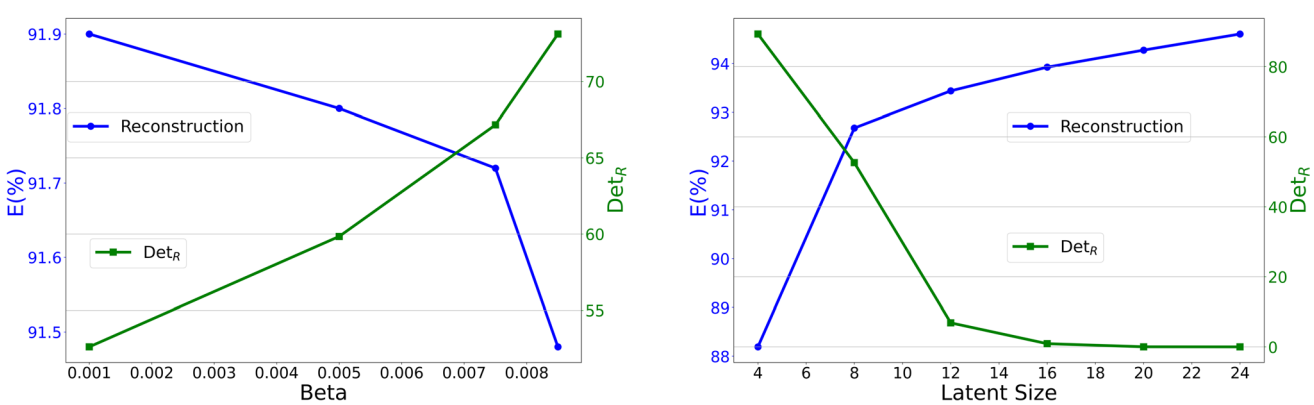

Fig. 8 Behavior of the reconstruction percentage (blue curve) and $\det(\mathrm{R})$ (green curve) as a function of beta with a latent dimension of 8 (left) and as a function of the dimension of the latent space with a β of 1.0×10^{-3} (right)

especially relevant in the context of β-VAEs, where higher β values promote a more structured and disentangled latent space by encouraging the posterior to approximate the isotropic prior $\mathcal{N}(0, I)$. In this setting, a systematic approach is required to identify the most informative latent modes for data reconstruction and meaningful variation capture.

The method proposed in Eivazi et al. [18] addresses this need by introducing a mode-ranking algorithm that yields physically relevant cumulative modes. The approach assumes alignment between the model's learned reference geometry and the latent space structure, a condition that holds particularly well in β-VAEs, where latent representations cluster around a central reference vector $z_i = 0_{i=1}^{d}$

This ranking method is employed in this work (Algorithm 1) to rank latent modes by reconstruction percentage $E(\%)$. A trained GCN-β-VAE encodes the high-dimensional input data M into its latent representation via the encoder $\mathcal{E} : M \mapsto \boldsymbol{\mu}, \boldsymbol{\sigma}$, using only the mean $\boldsymbol{z} = \hat{\boldsymbol{\mu}}$. To evaluate each mode's contribution, a modified latent vector is created by retaining only the ith component of $\boldsymbol{z}$ and zeroing the rest. This modified vector $\hat{\boldsymbol{z}}_i$ is decoded to produce a partial reconstruction $\hat{M}_i$, and the corresponding energy percentage $E_i(\%)$ quantifies the mode's influence. The mode yielding the highest energy is selected first. In subsequent steps, previously selected modes are preserved, and each remaining candidate is tested in combination to identify the next highest contributing mode. This iterative process continues until all modes are ranked, producing an ordered list of latent dimensions prioritized by their effectiveness in reconstructing the original data M. To illustrate how mode ranking affects reconstruction, the cumulative reconstruction percentage and the cumulative Chamfer error are evaluated across different latent space sizes in Fig. 9. Specifically, dimensions ranging from $d = 4$ to $d = 24$ are analyzed. The resulting curves display the typical pattern of cumulative explained variance observed in PCA, indicating that the latent space captures increasingly more variance as dimensionality grows.

Algorithm 1 Ranking GCN-β-VAE modes.

Require: Trained encoder $\mathcal{E}$ and decoder $\mathcal{D}$ of the CNN-βVAE; data $\mathbf{x}$
Ensure: $\mathbf{J}$, the vector containing the ranked mode indices
Initialize $\mathbf{J}$ as an empty vector
$\mathbf{d} \leftarrow$ (vector containing indices of all modes)
$\mu \leftarrow \mu, \sigma \leftarrow \mathcal{E}(\mathbf{M}^*)$
$z \leftarrow \mu$ ▷ Use the mean as the representative latent vector
for $j \in \mathbf{d}$ **do**
  Initialize $\mathbf{E}$ as empty
  for $i \in \mathbf{d}[\sim \mathbf{J}]$ **do**
    $\mathbf{I} \leftarrow [\, i, \mathbf{J} \,]$ ▷ Concatenate i with already-selected modes
    $\hat{\mathbf{z}}_i \leftarrow \mathbf{z}$, then zero out all latent variables *not in* $\mathbf{I}$
    $\hat{M}_i \leftarrow \mathcal{D}(\hat{\mathbf{z}}_i)$
    Append $E(M, \hat{M}_i)$ to $\mathbf{E}$ ▷ Compute energy/score
  end for
  $\mathbf{J} \leftarrow \mathbf{J} \cup \{\mathbf{d}[\sim \mathbf{J}][\arg\max(\mathbf{E})]\}$
end for
Output: $\mathbf{J}$, the ranked list of modes

Notably, except for the latent size 4, the curves remain consistently above the reference PCA curve for E and below for what concerns the chamfer error, which signifies that for the same number of modes, the VAE achieves superior reconstruction quality compared to PCA. This discrepancy is most pronounced in the first mode, namely the vector in which only the element at position j is nonzero; that is, $\{z_i = 0\}_{i=1, i\neq j}^{d} \cup \{z_j \neq 0\}$, indicating that a latent code near the reference geometry can achieve substantial reconstruction accuracy and shape fidelity compared to PCA. This observation suggests a form of inner compactness within the model's learned latent space. Furthermore, in Fig. 10, it is examined how this compactness evolves as the weight β on the KL-Divergence term increases. As expected, as β

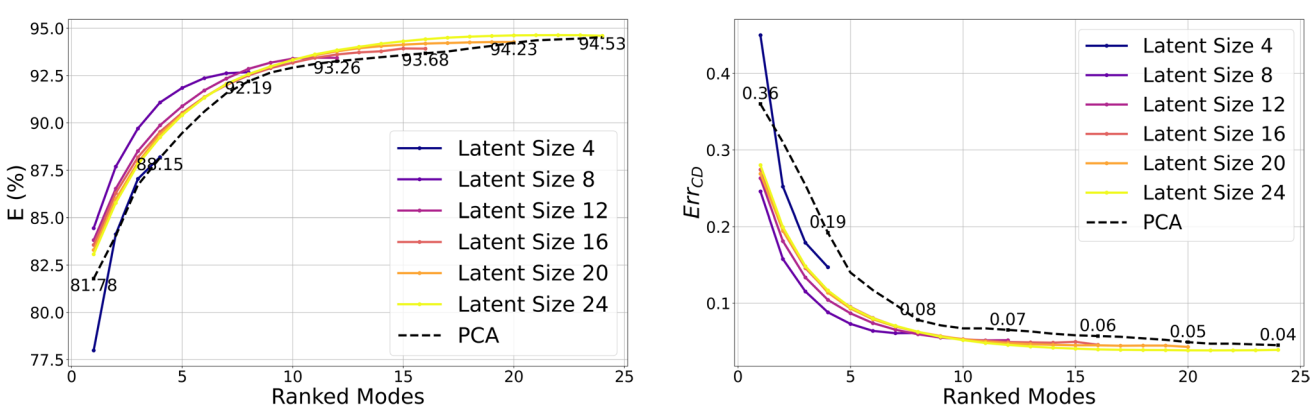

Fig. 9 Progression of cumulative reconstruction percentage (left) and Chamfer error (right) for latent dimensions ranging from $d = 4$ to $d = 24$. The curves resemble the typical pattern of cumulative explained variance in PCA. However, aside from the smallest latent space ($d = 4$), the GCN-β-VAE consistently outperforms PCA, achieving higher reconstruction accuracy and lower Chamfer error for each number of modes considered. These results emphasize the effectiveness of the learned mode ranking in improving reconstruction quality

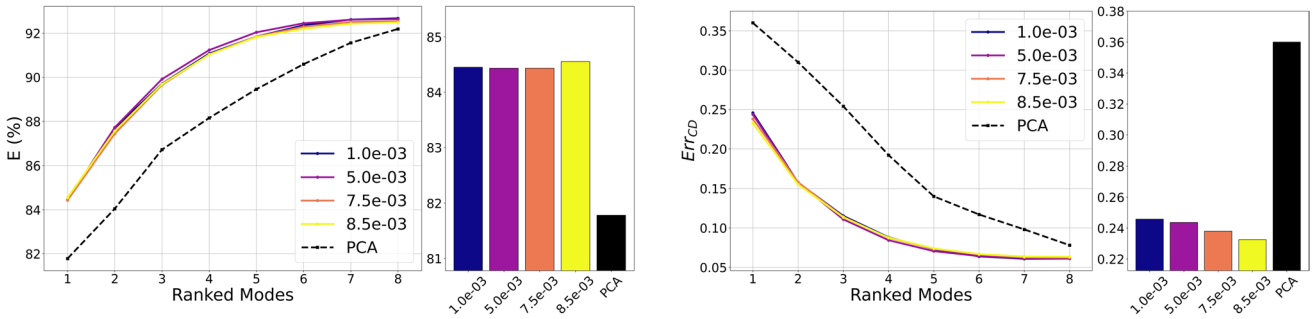

Fig. 10 Evolution of latent compactness as β increases, shown for both Chamfer error (right) and reconstruction percentage (left). The left panels depict the ranked latent modes as β varies from $1.0e-3$ to $8.5e-3$, while the right panels show the value of the first latent mode over the same range. As β increases, the reconstruction percentage E in the first mode slightly rises while the Chamfer error E_{rrCD} decreases, indicating that a stronger KL regularization term promotes a tighter, more structured latent space

increases, the reconstruction percentage E rises, while the Chamfer error E_{rrCD} decreases. Increasing β enforces a tighter alignment between the posterior and the standard Gaussian prior, effectively compressing and regularizing the latent space. These results reinforce the notion that stronger regularization via β improves compactness and structure within the latent space. Finally, Fig. 11 illustrates the reconstruction error, represented by the RCD values encoded in the mesh colors for one particular sample of the test set, the ground-truth mesh is also shown for direct comparison. While PCA achieves a reduction in the average integral distance as the number of modes increases, it fails to recover fine morphological details of the geometry, even when more modes are used. In contrast, the VAE not only preserves local geometric features with fewer modes but also progressively enhances the local accuracy of the reconstruction as

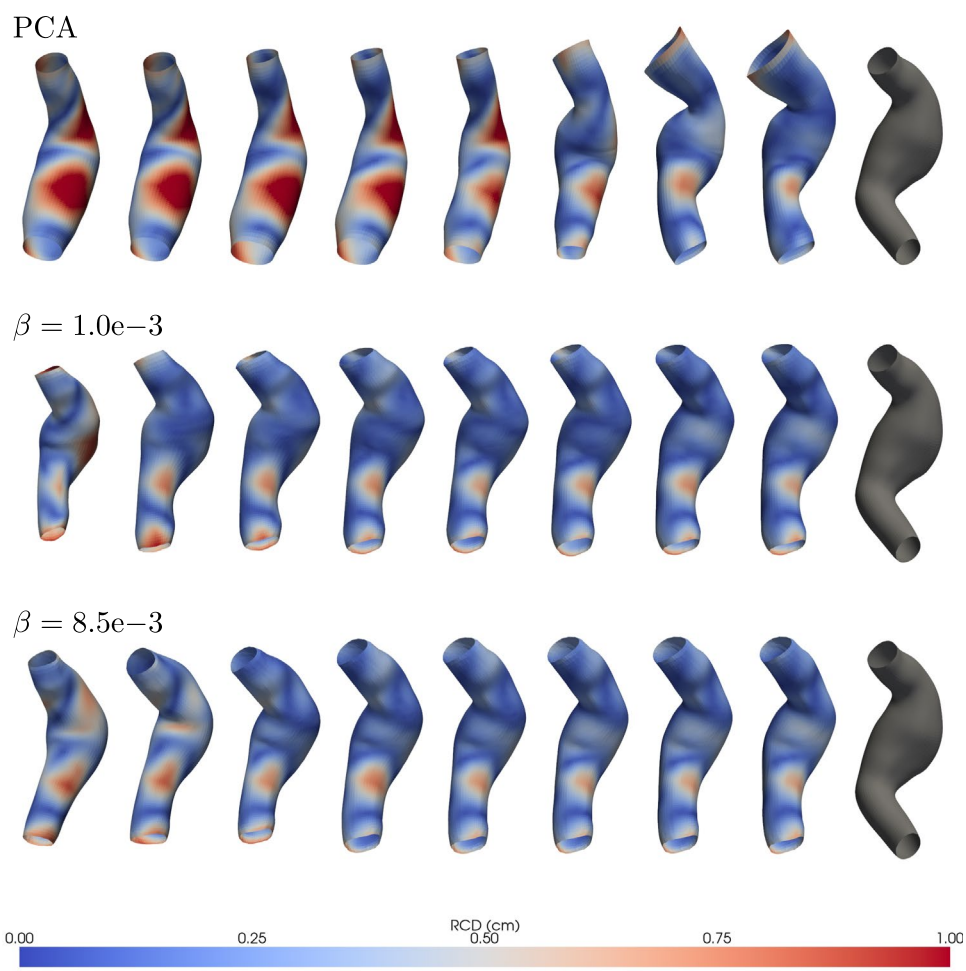

Fig. 11 Ranked mode reconstructions from PCA (top row), GCN-β-VAE with $\beta = 1.0 \times 10^{-3}$ (middle row), and GCN-β-VAE with $\beta = 8.5 \times 10^{-3}$ (bottom row). Each column shows the reconstructed shape from a single latent mode (or principal component), progressing from rank 1 to rank 8. Vertex-wise color coding is based on RCD (cm). The first modes of the GCN-β-VAE yield more compact deformations than PCA, which struggles especially in the higher order modes (far right), most notably along the boundary regions

additional modes are incorporated. With $\beta = 1\mathrm{e}{-3}$, the first mode alone yields an average error of 0.41 cm, which rapidly decreases to around 0.20 cm by the seventh mode. By contrast, PCA starts at 0.42 cm and remains above 0.3 cm until the sixth mode, ultimately dropping to 0.23 cm by the eighth mode. Increasing β to $8.5\mathrm{e}{-3}$ improves the VAE's initial mode (0.36 cm). Reconstruction via PCA exhibits pronounced artefacts, particularly in the neck and boundary regions, with these issues persisting even in the cumulative higher-order modes. In contrast, both GCN-β-VAEs provide more spatially compact and accurate reconstructions in early modes, with the first latent direction yielding a deformation significantly closer to the ground truth than PCA. This effect becomes more pronounced with increasing β, supporting the notion that stronger regularization induces greater compactness in the latent representation. Altogether, these findings demonstrate that heavier regularization (larger β) fosters a more compact and practical latent-space representation than PCA and lower-β settings.

3.3 In-sample generative experiments

This section outlines the experimental procedure used to generate AAA shapes with the GCN-β-VAE model. In Section 3.2, the focus was on evaluating the generalization capability of the GCN-β-VAE on unseen data. To this aim, both the regularization weight β and the latent dimension d are tuned to promote a latent space characterized by maximally independent factors. This independence is assessed through the analysis of the correlation matrix R of the latent variables, and the model configuration that maximizes $\det(\mathrm{R})$ indicating minimal correlation among latent dimensions is selected, provided that reconstruction accuracy remains adequate. With the optimal configuration identified as $\beta = 8.5 \times 10^{-3}$ and latent dimension $d = 8$, this model is subsequently applied to in-sample generative tasks. Specifically, in-sample generation is performed by either smoothly interpolating between two latent codes from the training dataset or by sampling from a Gaussian distribution centered around the latent code of a selected training sample. The interpolation procedure facilitates the generation of intermediate structural transitions, whereas Gaussian perturbations enable the modeling of localized variability surrounding instances. To validate anatomical plausibility, these generative results are shown alongside real ground-truth meshes in Figs. 12 and 13, enabling direct visual and quantitative comparison between synthetic and clinical anatomies. These experiments demonstrate the learned latent space capacity to synthesize new samples closely aligned with the training distribution, reinforcing the GCN-β-VAE's effectiveness in capturing and encoding meaningful latent structures within the training set. For generative purposes, only the mean of the latent distribution is used, such that $z_i = \mu_i$ instead of $z_i = \mu_i + \sigma_i \odot \epsilon$. During

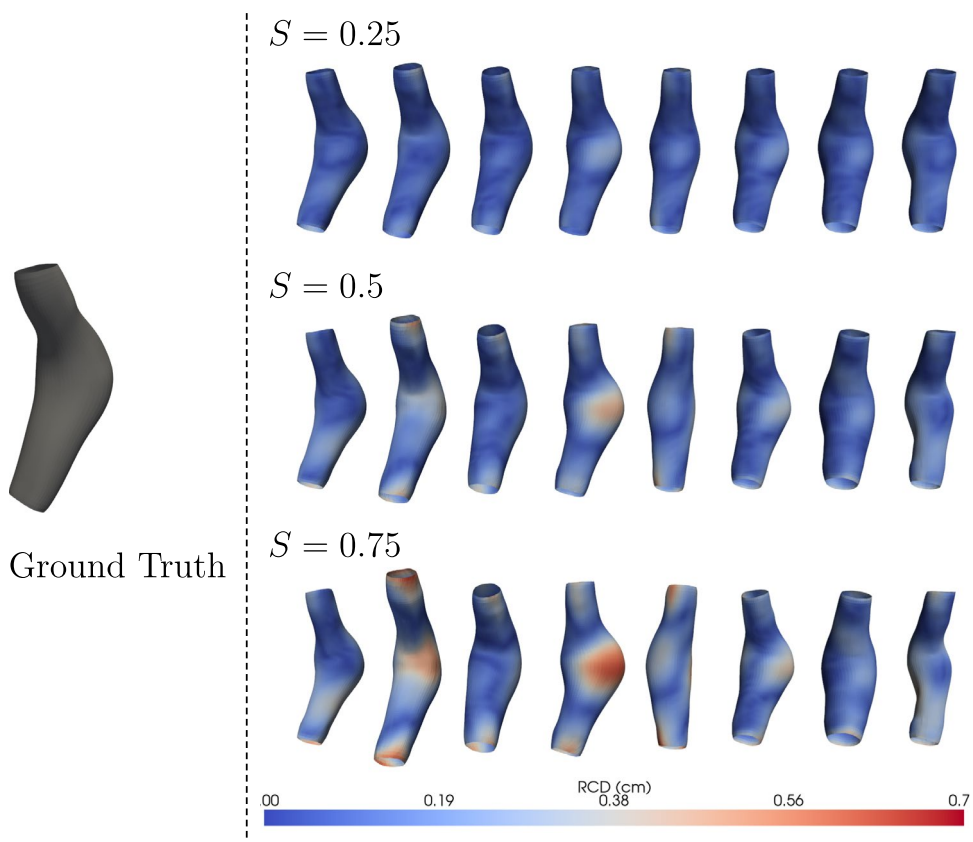

Fig. 12 Ground-truth aneurysm mesh (leftmost column) and RCD visualizations for extrapolated meshes generated with increasing levels of latent perturbation magnitude S. Rows correspond to $S = 0.25$, $S = 0.5$, and $S = 0.75$, from top to bottom. The RCD values are visualized as color maps over the surface of each extrapolated mesh, indicating the degree of geometric deviation from the ground truth. As S increases, deviations become more pronounced, with notable examples including aneurysm sac expansion (e.g., columns 4 and 6) and axial elongation (e.g., column 2). Despite these changes, global anatomical integrity of the abdominal aortic aneurysm (AAA) is preserved across all levels of perturbation

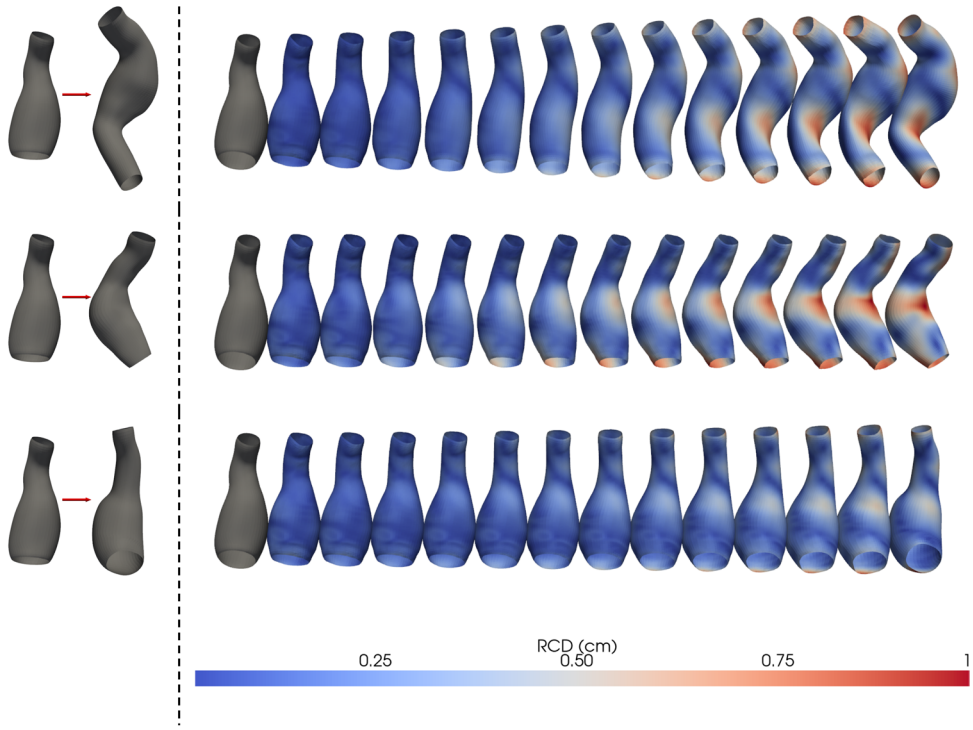

Fig. 13 Interpolation of latent codes. The left column shows the ground truth aneurysm meshes, while the right column presents the intermediate shapes generated by interpolating between latent representations. The color map encodes the root Chamfer distance (RCD), illustrating how the geometry transitions smoothly between the original and interpolated states

training, the stochastic term in the reparameterization trick serves to regularize and disentangle the latent space, but at generation it is omitted to ensure stable and reproducible reconstructions. Controlled variability is instead introduced through the generation strategies.

3.3.1 Extrapolation through Gaussian noise

An AAA mesh M is first embedded into a latent space by an encoder that maps it to a mean latent vector $\mu \in \mathbb{R}^d$. To explore shape variability while preserving anatomical realism, new mesh instances $\tilde{M}$ are generated by extrapolating from the latent mean using a controlled Gaussian perturbation:

$$z = \mu + S \cdot \Sigma, \quad \text{where} \quad \Sigma \sim \mathcal{N}(0, I),$$

where $S \in \mathbb{R}^+$ modulates the noise amplitude. Larger values of S produce more pronounced deviations from the ground truth. To ensure anatomical plausibility, extrapolation is constrained not to exceed the empirically observed ranges of the latent variables, as shown in Fig. 6 for the 8-dimensional latent space ($\beta = 8.5 \times 10^{-3}$). In practice, the generated samples were retrieved within three standard deviations of the observed latent distribution, which guarantees that the meshes remain realistic. Within these boundaries, the extrapolated meshes simulate plausible anatomical variations while maintaining topological consistency with the original AAA. The resulting extrapolated meshes $\tilde{M}$ simulate plausible anatomical variations while maintaining topological consistency with the original AAA.

As visualized in Fig. 12, the leftmost column shows the ground-truth mesh M, and the subsequent eight columns display extrapolated meshes $\tilde{M}_i$ generated with independent random seeds which serve as the control parameter for sample diversity. Each row corresponds to a different noise magnitude ($S = 0.25$, $S = 0.5$, and $S = 0.75$). The RCD is computed between each generated mesh and the ground truth and is shown as a heatmap over the mesh surfaces. As S increases, the extrapolated meshes exhibit greater geometric deviations. Notably, the fourth and sixth columns display a marked increase in aneurysm sac size, indicative of radial expansion, while the second column shows elongation along the vessel axis. These shape variations are consistent with clinically relevant deformations, underscoring the method's capacity to generate realistic anatomical diversity through controlled latent space perturbations.

3.3.2 Latent interpolation

In image data, interpolation is straightforward due to the inherent grid structure: pixels are spatially ordered and aligned, allowing for direct blending of intensity values. However, 3D point clouds, such as those representing AAAs, lack this spatial ordering [48]. Each point cloud is a set of points in $\mathbb{R}^3$ with no inherent order, making direct interpolation between two clouds ill-posed due to the absence of pointwise correspondence.

To overcome this issue, a learned latent space is exploited, in which shape representations are continuous and semantically structured. Given two AAA meshes M_1 and M_2, these meshes are first encoded into corresponding latent vectors μ_1 and μ_2 using the encoder. A sequence of intermediate latent codes is then defined by linear interpolation between these two vectors:

$$\mu(i) = \mu_1 + \frac{i}{n}(\mu_2 - \mu_1), \quad \text{for} \quad i = 0, 1, \ldots, n,$$

where n is the number of interpolation steps. Each interpolated latent vector $\mu(i)$ is then passed through the decoder to generate a mesh $\tilde{M}_i$:

This process yields a smooth sequence of mesh shapes transitioning from M to each respective target geometry, as shown in Fig. 13, where the leftmost column displays the ground-truth aneurysm geometry M, which is shared across all three interpolation rows as the common source shape, as well as the three target geometries. Each row corresponds to a different target shape, depicted in the rightmost column. The intermediate columns visualize decoded shapes from the interpolated latent vectors. The RCD is used to visualize deviations from the initial shape M through color-coded maps over each interpolated mesh. Interpolations are most reliable when the path between μ_1 and μ_2 passes through well-sampled regions of the latent space; traversals through sparse areas may yield less realistic intermediates. These visualizations confirm that interpolation in latent space produces anatomically plausible, continuous shape transformations.

4 Conclusion

In this work, a GCN-β-VAE framework capturing complex geometric variations in abdominal aortic aneurysm (AAA) morphology through a compact and disentangled latent representation is presented. Even when trained on a limited dataset, the model consistently outperforms classical techniques such as PCA on both the L_2 and Chamfer metrics, demonstrating robustness and generalization capabilities. By constraining the latent space via the β-VAE structure, clinically relevant morphological features are retained while redundancy among latent dimensions is minimized. This structured encoding enables smooth and interpretable shape synthesis via latent interpolation and extrapolation. The resulting synthetic AAAs preserve critical anatomical characteristics and exhibit clinically meaningful variations including sac diameter, neck angulation, and overall length. A physically plausible augmentation strategy (Procrustes RandAugment) further enriches the training data without compromising anatomical validity, enhancing performance on unseen cases, reducing reconstruction and correspondence errors, measured by RCD and L_2 norm, by up to one-fifth on a dataset of only 60 patient cases (see Table 1). Taken together, these elements transform a data-sparse clinical scenario into a source of virtual cohorts, benefiting device testing, haemodynamic simulation and in-silico trials.

Despite these advances, several limitations remain. The latent disentanglement achieved is partial, and some latent factors may still entangle correlated anatomical changes. Additionally, the proposed method is designed for watertight meshes. Therefore, in cases where the AAA surface mesh is open, a preprocessing step is required to close the mesh. This is addressed by creating geometric lids through radial basis function interpolation, allowing the resulting surfaces to be treated as watertight during training.

Future directions include extension to variable-topology meshes or point-cloud representations to capture broader anatomical diversity, semi-supervised learning using known key geometrical features (e.g., the centerline of the blood vessel), and large-scale validation on external datasets with wider anatomical variation beyond the current 60-patient cohort. In addition, a qualitative evaluation of richer latent space could improve the partial disentanglement observed in our framework. Moreover, incorporating supervised clinical labels could guide the latent space toward clinically actionable features. Finally, real-time generative tools may be developed for patient-specific surgical planning and in silico trials. In particular, synthetic AAAs generated by our framework could be integrated into clinical decision-making workflows, for example by supporting virtual stent-graft deployment and outcome simulation in EndoVascular Aneurysm Repair (EVAR) procedures.

Acknowledgements This project has received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation program (Grant No. 101039657, CARDIOTRIALS to F.V.). This work has been partially supported by Next Generation EU, Mission 4, Component 1, CUP D13C22001470001.

Declarations

Competing Interests The authors declare that they have no competing financial interests or personal relationships that could have appeared to influence the work presented in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Pezoulas VC, Zaridis DI, Mylona E, Androutsos C, Apostolidis K, Tachos NS, Fotiadis DI (2024) Synthetic data generation methods in healthcare: a review on open-source tools and methods. Comput Struct Biotechnol J 23:2892–2910. https://doi.org/10.1016/j.csbj.2024.07.005
2. Eigenschink P, Reutterer T, Vamosi S, Vamosi R, Sun C, Kalcher K (2023) Deep generative models for synthetic data: a survey. IEEE Access 11:47304–47320. https://doi.org/10.1109/ACCESS.2023.3275134
3. Tucker A, Wang Z, Rotalinti Y, Myles P (2020) Generating high-fidelity synthetic patient data for assessing machine learning healthcare software. Npj Digit Med 3(1):147. https://doi.org/10.1038/s41746-020-00353-9
4. Lucini F (2021) The real deal about synthetic data. Data, AI, & Machine Learning. Accessed 25 03 2025
5. Wang Y, Seguro F, Kao E, Zhang Y, Faraji F, Zhu C, Haraldsson H, Hope M, Saloner D, Liu J (2017) Segmentation of lumen and outer wall of abdominal aortic aneurysms from 3d black-blood mri with a registration based geodesic active contour model. Med Image Anal 40. https://doi.org/10.1016/j.media.2017.05.005
6. Whitaker SC (2001) Imaging of abdominal aortic aneurysm before and after endoluminal stent-graft repair. Eur J Radiol 39(1):3–15. https://doi.org/10.1016/S0720-048X(01)00337-0
7. Czermak BV, Fraedrich G, Schocke MF et al (2001) Serial ct volume measurements after endovascular aortic aneurysm repair. J Endovasc Ther 8(4):380–389. https://doi.org/10.1177/152660280100800407
8. Becker C, Bülow T, Gombert A, Kalder J, Keschenau PR (2022) Infrarenal remains infrarenal—evar suitability of small aaa is rarely compromised despite morphological changes during surveillance. J Clin Med 11(18):5319. https://doi.org/10.3390/jcm11185319
9. Bourlard H, Kamp Y (1988) Auto-association by multilayer perceptrons and singular value decomposition. Biol Cybern 59(4):291–294. https://doi.org/10.1007/BF00332918
10. Rolínek M, Zietlow D, Martius G (2019) Variational autoencoders pursue pca directions (by accident). In: 2019 IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR), pp 12398–12407. https://doi.org/10.1109/CVPR.2019.01269
11. Chen RTQ, Li X, Grosse R, Duvenaud D (2019) Isolating sources of disentanglement in variational autoencoders. arXiv:1802.04942
12. Dai B, Wang Y, Aston J, Hua G, Wipf D (2019) Hidden talents of the variational autoencoder. arXiv:1706.05148
13. Baur C, Denner S, Wiestler B, Navab N, Albarqouni S (2021) Autoencoders for unsupervised anomaly segmentation in brain mr images: a comparative study. Med Image Anal 69:101952. https://doi.org/10.1016/j.media.2020.101952
14. Jiang Z-H, Wu Q, Chen K, Zhang J (2019) Disentangled representation learning for 3d face shape. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)
15. Ge L, Ren Z, Li Y, Xue Z, Wang Y, Cai J, Yuan J (2019) 3d hand shape and pose estimation from a single rgb image. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)
16. Zhao L, Peng X, Tian Y, Kapadia M, Metaxas DN (2019) Semantic graph convolutional networks for 3d human pose regression. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)
17. Zhang Z, Yu C, Li H, Sun J, Liu F (2020) Learning distribution independent latent representation for 3d face disentanglement. In: 2020 International conference on 3D Vision (3DV), pp 848–857. https://doi.org/10.1109/3DV50981.2020.00095
18. Eivazi H, Le Clainche S, Hoyas S, Vinuesa R (2022) Towards extraction of orthogonal and parsimonious non-linear modes from turbulent flows. Expert Syst Appl 202:117038. https://doi.org/10.1016/j.eswa.2022.117038
19. Ranjan A, Bolkart T, Sanyal S, Black MJ (2018) Generating 3d faces using convolutional mesh autoencoders. In: Proceedings of the European Conference on Computer Vision (ECCV)
20. Zou K, Faisan S, Heitz F, Epain M, Croisille P, Fanton L, Valette S (2023) Disentangled representations: towards interpretation of sex determination from hip bone. Vis Comput 39(12):6673–6687. https://doi.org/10.1007/s00371-022-02755-0
21. Bonazzola R, Ferrante E, Ravikumar N, Xia Y, Keavney B, Plein S, Syeda-Mahmood T, Frangi AF (2023) Unsupervised ensemble-based phenotyping helps enhance the discoverability of genes related to heart morphology. arXiv:2301.02916
22. Moll FL, Powell JT, Fraedrich G, Verzini F, Haulon S, Waltham M, van Herwaarden JA, Holt PJE, van Keulen JW, Rantner B, Schlösser FJV, Setacci F, Ricco J-B (2011) Management of abdominal aortic aneurysms clinical practice guidelines of the European society for vascular surgery. Eur J Vasc Endovasc Surg 41:1–58. https://doi.org/10.1016/j.ejvs.2010.09.011. Managemen t of Abdominal Aortic Aneurysms. Clinical Practice Guidelines of the European Society for Vascular Surgery
23. Jin Y, Pepe A, Li J, Gsaxner C, Zhao F-h, Pomykala KL, Kleesiek J, Frangi AF, Egger J (2023) AI-based aortic vessel tree segmentation for cardiovascular diseases treatment: status quo. arXiv:2108.02998
24. Caradu C, Spampinato B, Vrancianu AM, Bérard X, Ducasse E (2021) Fully automatic volume segmentation of infrarenal abdominal aortic aneurysm computed tomography images with deep learning approaches versus physician controlled manual segmentation. J Vasc Surg 74(1):246–2566. https://doi.org/10.1016/j.jvs.2020.11.036
25. Riffaud S, Ravon G, Allard T, Bernard F, Iollo A, Caradu C (2022) Automatic branch detection of the arterial system from abdominal aortic segmentation. Med Biol Eng Comput 60(9):2639–2654. https://doi.org/10.1007/s11517-022-02603-2. (**Epub 2022 Jul 7**)
26. Ravon G, Bernard F, Iollo A, Caradu C (2022) Automatic rigid registration of aortic aneurysm arterial system. Working paper or preprint. https://inria.hal.science/hal-03600571
27. Lorensen WE, Cline HE (1987) Marching cubes: a high resolution 3d surface construction algorithm. SIGGRAPH Comput Graph 21(4):163–169. https://doi.org/10.1145/37402.37422
28. Piccinelli M, Veneziani A, Steinman DA, Remuzzi A, Antiga L (2009) A framework for geometric analysis of vascular structures: application to cerebral aneurysms. IEEE Trans Med Imaging 28(8):1141–1155
29. Antiga L, Steinman DA (2004) Robust and objective decomposition and mapping of bifurcating vessels. IEEE Trans Med Imaging 23(6):704–713
30. Saccaro L, Ravon G, Nastorg M, Philibert T, Bernard F, Iollo A (2024) Automatic parameterisation and registration of abdominal aortic aneurysm. In: Skalli W, Laporte S, Benoit A (eds) Computer methods in biomechanics and biomedical engineering II. Springer, Cham, pp 169–176
31. Kingma DP, Welling M (2022) Auto-encoding variational Bayes
32. Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, Lerchner A (2017) beta-VAE: learning basic visual concepts with a constrained variational framework. In: International conference on learning representations. https://openreview.net/forum?id=Sy2fzU9gl
33. Wang N, Zhang Y, Li Z, Fu Y, Liu W, Jiang Y-G (2018) Pixel2Mesh: generating 3D mesh models from single RGB images. arXiv:1804.01654

34. Choi H, Moon G, Lee KM (2021) Pose2Mesh: graph convolutional network for 3D human pose and Mesh recovery from a 2D human pose. arXiv:2008.09047
35. Defferrard M, Bresson X, Vandergheynst P (2017) Convolutional neural networks on graphs with fast localized spectral filtering. arXiv:1606.09375
36. Garland M, Heckbert PS (1997) Surface simplification using quadric error metrics. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques. SIGGRAPH '97, pp 209–216. ACM Press/Addison-Wesley Publishing Co., USA. https://doi.org/10.1145/258734.258849
37. Cubuk ED, Zoph B, Mane D, Vasudevan V, Le QV (2019) Autoaugment: learning augmentation strategies from data. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)
38. Cubuk ED, Zoph B, Shlens J, Le QV (2019) RandAugment: practical automated data augmentation with a reduced search space. arXiv:1909.13719
39. Gower JC (1975) Generalized procrustes analysis. Psychometrika 40(1):33–51. https://doi.org/10.1007/BF02291478
40. Bookstein FL (1992) Morphometric tools for landmark data: geometry and biology. Cambridge University Press
41. Bernardes E, Viollet S (2022) Quaternion to Euler angles conversion: a direct, general and computationally efficient method. PLoS ONE 17(11):0276302. https://doi.org/10.1371/journal.pone.0276302
42. Guo Y, Wang H, Hu Q, Liu H, Liu L, Bennamoun M (2021) Deep learning for 3d point clouds: a survey. IEEE Trans Pattern Anal Mach Intell 43(12):4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434
43. Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H (eds) Advances in neural information processing systems 33:pp 6840–6851. Curran Associates, Inc
44. Wu T, Pan L, Zhang J, Wang T, Liu Z, Lin D (2021) Density-aware chamfer distance as a comprehensive metric for point cloud completion. arXiv:2111.12702
45. Montanaro A, Valsesia D, Magli E (2022) Rethinking the compositionality of point clouds through regularization in the hyperbolic space. arXiv:2209.10318
46. Burgess CP, Higgins I, Pal A, Matthey L, Watters N, Desjardins G, Lerchner A (2018) Understanding disentangling in β-VAE. arXiv:1804.03599
47. Kim H, Mnih A (2019) Disentangling by factorising. arXiv:1802.05983
48. Mao J, Wang X, Li H (2019) Interpolated convolutional networks for 3D point cloud understanding. arXiv:1908.04512

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Francesco Fabbri is a Ph.D. candidate in Mathematics at Gran Sasso Science Institute, researching machine learning-based generative models for nonlinear biological tissue generation and statistical shape modeling. He earned his MSc in Statistical and Computational Physics from Tor Vergata, Rome, Italy.

Martino Andrea Scarpolini is Assistant Professor at Gran Sasso Science Institute, specializing in cardiac hemodynamics through high-fidelity numerical simulations and fluid-structure interaction. His research integrates turbulence modeling, GPU-based computations, and machine learning for medical imaging.

Angelo Iollo is Professor of Applied Mathematics at the University of Bordeaux, previously leading Inria's Memphis team. His work spans computational mathematics, national and EU project leadership, and entrepreneurial ventures, including co-founding startups Nurea and Optimad Engineering.

Francesco Viola is Professor of Fluid Mechanics at Gran Sasso Science Institute. His research centers on multi-physics computational modeling of the human heart, fluid-structure interaction, hemodynamics, and fluid mechanics instabilities.

Francesco Tudisco is Associate Professor in Machine Learning at the University of Edinburgh and affiliated with Gran Sasso Science Institute. His research blends machine learning and scientific computing, with a focus on physics-inspired deep learning models, deep learning theory, and optimization.