Variational 3D Mesh Generation of Man-Made Objects

George Fahim
Dept. of Information Technology
Faculty of Computers and Information
Menofia University, Egypt
Multimedia and Internet Dept.
International Academy for Engineering and Media Science
Giza, Egypt
george.fahim@iams.edu.eg

Khalid Amin
Dept. of Information Technology
Faculty of Computers and Information
Menofia University, Egypt
k.amin@ci.menofia.edu.eg

Sameh Zarif
Dept. of Information Technology
Faculty of Computers and Information
Menofia University, Egypt
sameh.shenoda@ci.menofia.edu.eg

***Abstract*—Data-driven 3D shape analysis, reconstruction, and generation is an active research topic that finds many useful applications in the fields of computer games, computer graphics, and augmented/virtual reality. Many of the previous mesh-based generative approaches work on natural shapes such as human faces and bodies and little work targets man-made objects. This work proposes a generative probabilistic framework for 3D man-made mesh shapes. Specifically, it proposes a Variational Autoencoder that works directly on mesh vertices and encodes meshes into a probabilistic, smooth, and traversable latent space that can be sampled after training and decoded to generate novel and plausible shapes. Extensive experiments show the representational power of the proposed framework and the underlying latent space. Operations such as random sample generation, linear interpolation, and shape arithmetic can be performed using the proposed method and produce plausible results. An additional advantage of the proposed framework is that it learns to produce a disentangled shape representation which gives finer control over the generated mesh and allows generating shapes with specific qualities without losing the reconstruction power of the autoencoder.**

***Index Terms*—geometric deep learning, representation learning, 3D mesh generation, variational autoencoding, shape interpolation, disentangled shape representation**

I. INTRODUCTION

The field of data-driven 3D shape analysis and generation has gained a considerable momentum in the recent years. This can be attributed to many contributing factors such as the recent advances made in 3D and geometrical deep learning [1], the proposal of neural architectures that can process 3D shapes [2], the recent availability of large 3D shapes datasets [3]–[6], and the advances made in compute capabilities. However, there are many challenges that remain unsolved, thus require further investigations. Among these challenging tasks is the task of using the 3D mesh representation in Single-View Reconstruction (SVR) and 3D shape analysis and generation.

There are different shape representations that can be used in the tasks of data-driven 3D shape analysis and generation. The most commonly used shape representations are regular voxel grids, point clouds, meshes, and implicit surface functions [2]. Each of these shape representations has its own merits and demerits. Voxel grids are well structured and easy to process using neural networks, but are computationally and memory expensive. Point clouds are more memory efficient, but lack connectivity or surface information. Meshes have connectivity and surface information, but are difficult to handle in neural networks. Implicit functions can produce shapes of any arbitrary topology, but require post-processing.

This work focuses on the generation of man-made 3D shapes using the mesh representation. Despite the difficulty of using 3D meshes in data-driven tasks, they are considered one of the most versatile and efficient shape representations. Using data-driven methods to automatically generate 3D meshes helps in populating 3D assets easily. These assets can be used in computer games, 3D animation films, virtual reality, and augmented reality [7]. Meshes can also be used in the field of machine vision and robotics to allow for 3D reasoning in navigation, planning, and grasping tasks.

The difficulty of using meshes in data-driven 3D shape analysis and generation arises from their unstructured and arbitrary nature. A 3D mesh is composed of a number of vertices and edges connecting them. Connected edges form polygonal faces that represent the surface of the 3D object. Faces can be made up of 3 connected edges (triangular faces), 4 connected edges (quadrilateral faces), or 5 or more connected edges (n-gons). Additionally, a 3D mesh can go beyond genus0 and have one or more holes in them. This arbitrariness in mesh parameterization makes it a difficult representation to be processed using neural networks which are better suited to work in the Euclidean domain where the data is well structured.

To overcome this difficulty, researchers proposed different methods for working with 3D meshes such as: working with intermediate representations like geometry images [8] and displacement maps [9], deforming a template mesh [10], [11] in SVR tasks, or through working with axis-aligned meshes with the same topology [12]–[14] in generative and analysis tasks. Many of these methods have been facilitated by the introduction of graph-based convolutional networks [15].

This paper proposes a generative framework for the probabilistic generation of man-made 3D shapes based on the 3D mesh representation. It specifically introduces a Variational Autoencoder (VAE) [16] that works on mesh vertices directly, captures shape features at different levels of details, and encodes shapes into a compressed, disentangled, smooth, and traversable probabilistic latent space. This allows shape decoding through picking samples from the latent space and generating shapes which can be novel, smooth, and plausible.

II. RELATED WORK

The earliest methods in data-driven shape generation were proposed in the context of SVR. These methods aimed at reconstructing and generating shapes given a single image as an input and used the voxel representation to describe the generated shapes. The 3D-GAN [17] model extended the 2D GAN [18] to work with 3D data, while the TL-Embedding network [19] used an autoencoder to reconstruct and generate 3D shapes in voxel format. In contrast to our proposed method, the outputs of these methods are coarse and lack fine details. Mandikal et al. [20] proposed a VAE to reconstruct and generate point clouds which lack connectivity as opposed to our mesh-based method which inherently leverages edge connectivity. When it comes to using the mesh representation, Henderson and Ferrari [21] proposed a 2D supervised method that uses a VAE to generate meshes by using a fixed nonlearned mesh parameterization function. On the other hand, Pavllo et al. [9] proposed using a GAN to generate texture and displacement maps as an intermediate representation to generate textured 3D meshes through a 2D supervised training paradigm. Fahim et al. [22] proposed an autoencoder that takes axis-aligned 3D meshes with the same topology to generate novel shapes using a 3D supervised training paradigm. The autoencoder of this method is deterministic which restricts its generative power to produce varied shapes. In contrast and to increase the generative capability of the proposed method, a probabilistic approach is employed instead.

In the context of shape analysis and generation, the input to the network is a 3D shape and the output can either be a reconstruction of the shape, shape semantic segmentation, shape classification, or shape correspondence. Gao et al. [23] introduced a fully connected VAE that works on mesh features for tasks like mesh generation, interpolation, and deformation. In [13] and [14] the researchers introduced pooling and unpooling operations within a graph-based convolutional VAE. Their proposed convolutions operate in the spectral domain, and their pooling/unpooling operations rely on mesh simplification using edge contraction. The upsampling in [13] relies on using the barycentric coordinates in triangles of the coarse mesh to recover the lost vertices by interpolation, while in [14] aggregates local information by keeping record of the simplification process which allows reversing the process to perform unpooling. Lim et al. [24] proposed an alternative convolution operator on meshes by serializing vertex neighborhoods using a spiral sequence. They proposed two architectures, one based on Long Short-Term Memory (LSTM) cells to handle variable-sized sequences, and the other based on fully connected layers to handle fixed-sized sequences. MeshCNN proposed by Hanocka et al. [25] proposed dynamic task-specific pooling and unpooling operations. In general, these methods are aimed at natural shapes, while the proposed work is adapted to handle man-made objects.

Architectures that focus mainly on man-made mesh analysis and generation include SDM-NET [26] and PolyGen [7]. SDM-NET [26] is a generative model that generates meshes composed of deformable parts using a two-level VAE to generate part-based shape structure and geometry. This method requires part-level annotations and applies a set of structural constraints to generate plausible and correctly structured shapes. PolyGen [7], on the other hand, is an autoregressive generative model that is able to generate variable-sized meshes by generating mesh vertices then generating faces conditioned on these vertices using a Transformer-based architecture.

III. PROPOSED METHOD

The proposed framework is a convolutional VAE that consists of an encoder and a decoder. The encoder takes a 3D mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ —where $\mathcal{V}, \mathcal{E}, \mathcal{F}$ are the mesh's vertices, edges, and faces respectively —as an input, and learns to encode it into a set of probability distributions, thus learning a smooth probabilistic latent space. The decoder, on the other hand, is fed a latent vector sampled from the latent space and decodes it back into a 3D mesh. The overall architecture of the proposed framework is illustrated in Fig. 1. The encoder and the decoder of the proposed VAE are built on two main operations: an intrinsic graph convolutional operator, and a mesh simplification operation that allows for the application of pooling and unpooling operations by performing mesh decimation through edge contraction and creating a downtransform matrix and an up-transform matrix which are used in the actual pooling and unpooling operations respectively.

A. Graph Convolution

The proposed method adopts the spiral graph convolution operator proposed in [24] and further simplified in [12]. The main aim of this spiral convolution operator is to inherit the simplicity of the standard convolution operation used in the structured Euclidean domain (such as in CNNs) and adapt it to the geometrical and unstructured non-Euclidean domain. This is achieved by the introduction of a serialization process in which the neighborhood of each vertex is serialized in a spiral sequence with a fixed length. This fixed size spiral sequence mimics the kernel used in standard convolutions. The benefit of this adaptation is that it eliminates the requirement for the tedious neighborhood aggregation process that is usually required in graph convolutions. The spiral graph convolution is thus defined as:

$$\mathbf{x}_i^{(k)} = \gamma^{(k)}\left(\|_{j \in S(i,l)}\, \mathbf{x}_j^{(k-1)}\right), \qquad (1)$$

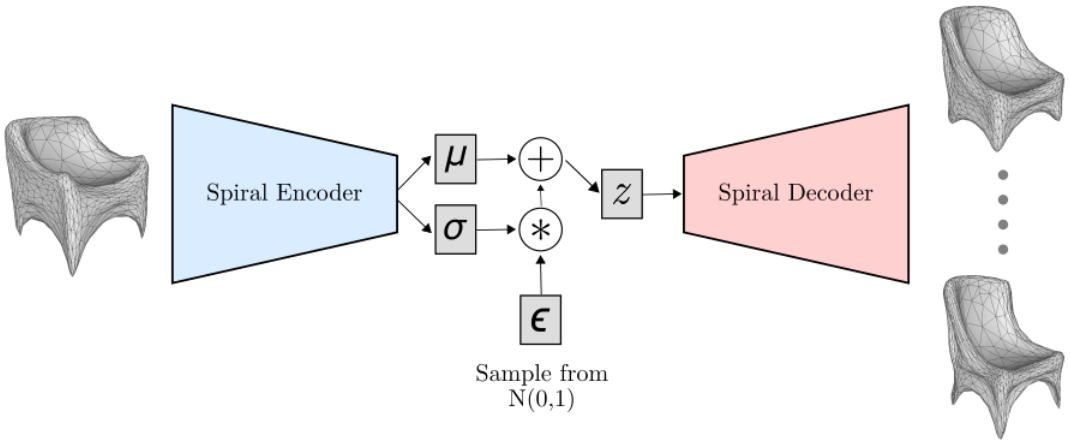

Fig. 1. The overall architecture of the proposed framework. A 3D mesh is fed into an encoder that performs spiral convolutions [12] and pooling operations [13] to generate a probabilistic representation of the shape. Sampling is performed to get a latent code that is fed into the decoder to reconstruct the input shape and produce novel plausible variants of it.

where $\mathbf{x}_i^{(k)}$ denotes node's i features in layer k, $S(i,l)$ is the spiral sequence around node i with length l, $||$ denotes the concatenation of the features of the nodes in the spiral sequence, and $\gamma^{(k)}$ is a linear transformation.

B. Pooling and Unpooling

In order to build a mesh hierarchy with different levels of details, the proposed framework adopts the mesh decimation operation based on edge contraction proposed in [13]. This allows the network to learn shape features at different scales which can help learning both global and local features. The original mesh is simplified through iteratively collapsing vertex pairs that minimize the quadric error [27] thus simplifying the mesh while maintaining its overall shape. During the process of mesh decimation, both the retained and the discarded vertices are kept track of in a down-transform matrix and are also used to build the up-transform matrix that does the opposite up-sampling operation. The pooling operation is thus performed by multiplying the mesh with the downtransform matrix and the unpooling operation is performed by multiplying the down-sampled mesh with the corresponding up-transform matrix.

C. Network Structure

The encoder of the proposed VAE is composed of four convolutional blocks each block is made up of a spiral convolution layer followed by a pooling layer and an Exponential Linear Unit (ELU) activation. The last layer of the encoder is a linear layer that outputs mean and standard deviation values. The decoder's structure mirrors the encoder's layers. However, the up-transform matrices are used instead of the down-transform matrices to perform unpooling instead of pooling. An additional spiral convolution layer is added to the end of the decoder with an output dimension of 3 to output the coordinates of the vertices of the reconstructed 3D mesh.

The loss function used during training the proposed model is made up of two terms. The first term is the reconstruction loss that penalizes the inaccurate predictions of the model when compared to the input mesh. Mean Squared Error (MSE) is used as the reconstruction loss. The second term is the generative term which uses the Kullback–Leibler (KL) divergence to encourage the learned posterior distribution to be similar to the true prior distribution. The total loss function is thus defined as:

$$L = \alpha \frac{1}{M} \sum_{i=1}^{M} \|X_i - \hat{X}_i\|_2 + \beta D_{KL}\big(q(z \mid X) \parallel p(z)\big), \qquad (2)$$

where X_i and $\hat{X}_i$ are the predicted and input meshes of the i^{th} 3D shape, M is the total number of 3D shapes, z is the latent code, $q(z \mid X)$ is the posterior probability, $p(z)$ is the prior probability, D_{KL} is the KL-divergence, and α and β are hyperparameter values balancing the two terms of the loss function.

IV. EXPERIMENTS AND RESULTS

A. Setup and Implementation Details

1) Dataset: The proposed framework requires aligned 3D meshes with the same topology. Such requirement is fulfilled in natural shapes datasets such as FAUST [5] and 4DFAB [6]. However, and to the authors' best knowledge, none of the existing datasets of man-made 3D objects conforms to this requirement. The unavailability of such data is overcome by starting with four categories from the ShapeNet dataset [3], namely the chair, plane, table, and sofa categories, and using the mesh reconstruction module in [22] to reconstruct the shapes in these categories. This SVR module [22] takes as input the renderings of the shapes in the dataset, encodes them, and learns to decode them back into 3D shapes by deforming a template spherical mesh into their respective shapes. The reconstructed meshes are aligned and with the same topology since they all result from deforming the initial template sphere. The training/testing split of the reconstructed dataset was set to 75/25.

TABLE I.
QUANTITATIVE COMPARISON OF THE RECONSTRUCTION RESULTS ON UNSEEN DATA BETWEEN THE PROPOSED VAE AND THE MESH AUTOENCODER PROPOSED IN [22] MEASURED USING THE ROOT MEAN SQUARE (RMS) ERROR METRIC REPORTED IN CENTIMETERS.

	Mesh autoencoder [22]		**Proposed mesh VAE**	
	RMS error	***Median error***	***RMS error***	***Median error***
chair	0.0567	0.0427	0.0361	0.0293
plane	0.0183	0.0119	0.0225	0.0163
table	0.0761	0.0533	0.0514	0.0403
sofa	0.0155	0.0117	0.0172	0.0140

2) Preprocessing: Each of the meshes in the reconstructed ShapeNet dataset goes through two preprocessing steps. The first one is to generate the decimated versions of the mesh and create the related down-transform and corresponding uptransform matrices that are required for the pooling/unpooling operations. The second step is to extract the spiral sequences for all the vertices in all the versions (the original and decimated ones) of the meshes in the dataset.

3) Implementation Details: The proposed model is implemented using PyTorch and trained using Adam optimizer [28] for 500 epochs with an initial learning rate of 5e−4, learning rate decay of 0.99, and a batch size of 10. The latent space dimension is set to 256, the decimation factor is 2, and the spiral sequence length is 9.

B. Reconstruction Evaluation

The first aspect of evaluation is to test the reconstruction accuracy of the proposed method. Table I shows a quantitative comparison of the reconstruction results on unseen data between the proposed method and the mesh autoencoder, proposed in [22], measured using the Root Mean Square (RMS) error metric reported in centimeters. This method was chosen for comparison since it also tackles the same problem of autoencoding man-made objects and uses the same convolution and pooling operations which makes comparison fair between the two methods. The proposed VAE outperforms the mesh autoencoder [22] in two out of four categories, while produces comparable results in the other two categories. It is worth mentioning that the mesh autoencoder [22] is deterministic and its sole objective is the accurate reconstruction of input mesh. However, because of the probabilistic nature of the proposed framework there is a tension and subsequently a trade-off between the reconstruction accuracy and the measure of similarity between the prior and posterior distributions – which is enforced by the KL-divergence term in (2). Qualitative reconstruction comparison with [22] is shown in Fig. 2. It visually shows that the proposed method outperforms [22] in the chair and table categories and produces indistinguishable reconstruction results in the plane and sofa categories.

C. Generation of Novel Shapes

Another aspect of evaluation is to evaluate the underlying latent space. The first common way to test the representational power of the latent space is to randomly pick samples, decode

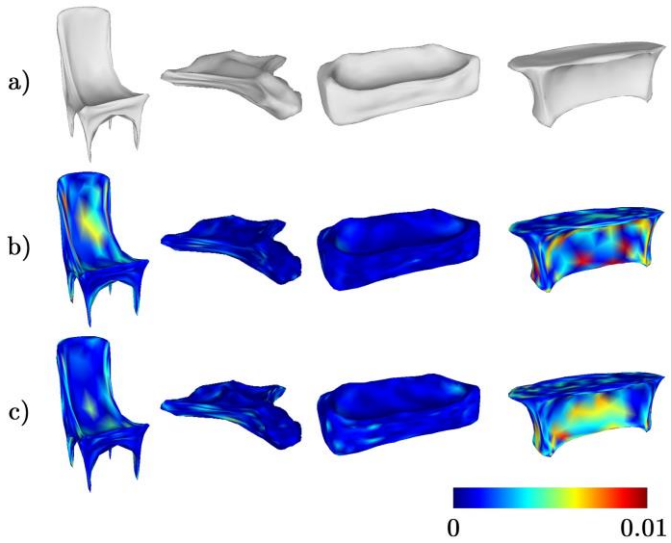

Fig. 2. Qualitative comparison of the reconstruction results on unseen data between the proposed VAE and the mesh autoencoder proposed in [22]. a) ground truth meshes, b) mesh autoencoder results [22], c) The proposed mesh VAE results. Reconstruction errors are color coded and red color saturates at reconstruction error of 0.01 cm.

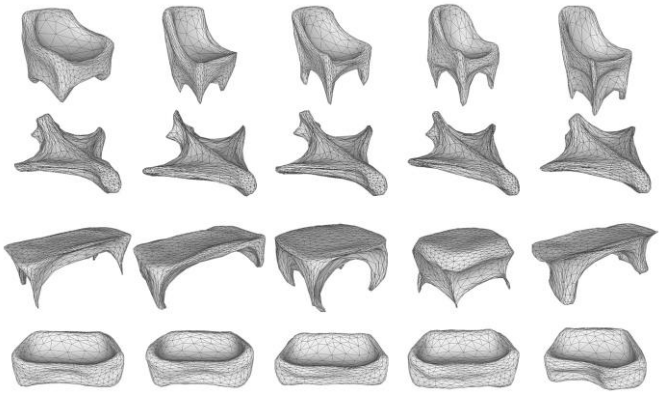

Fig. 3. Randomly generated new shapes using the proposed framework. Generation of these novel shapes is achieved by randomly sampling the latent space and feeding the sampled code to the proposed VAE's decoder.

and visually inspect the generated meshes for plausibility and variability. The sampling is done from the standard normal distribution $z \sim N(0, I)$, and the sampled codes are fed to the trained decoder to generate novel shapes. Fig. 3 shows the results of generating novel shapes, which shows the representational ability of the latent space to produce meshes that are both plausible and varied in shape.

D. Mesh Interpolation

The proposed method is also capable of interpolating between two meshes, which also confirms the generative capability of the proposed method. Instead of using meshes from the test set to investigate the interpolation results, first, two latent codes are sampled, and then a linear interpolation is performed between the two sampled codes, and finally all the codes are fed into the decoder to generate meshes corresponding to the input

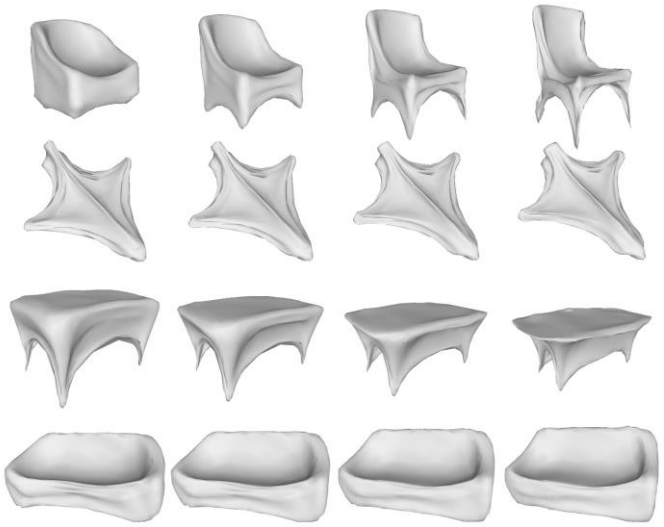

Fig. 4. Mesh interpolation results. The right- and left-most meshes are randomly sampled then their codes are linearly interpolated to generate the two intermediate results in the middle.

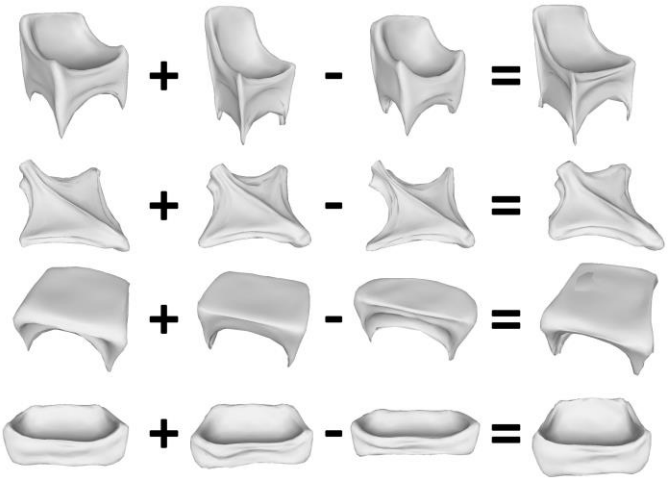

Fig. 5. Examples showing the application of arithmetic operations on latent codes through depicting their respective decoded meshes. Arithmetic operations prove that the latent space incorporates some sort of semantic structure that allow for the production of interpretable results.

codes. Fig. 4 shows the results of the linear interpolation operation which proves that the latent space is smooth and traversable.

E. Arithmetic Operations on Latent Codes

It has been observed that probabilistic generative models learn to imbue some sort of semantic structure into their learned latent space [17], [19]. This has been evidenced by the application of basic arithmetic operations on codes sampled from the latent space which usually produce structurally interpretable results. While the results are not as easily interpretable as the results of similar operations performed on word embedding models such as word2vec [29], however, they show whether the learned latent space has any embedded semantic knowledge or not. Fig. 5 shows the results of applying arithmetic operations on latent codes sampled from the latent space of the proposed method. These results show that the latent space incorporated some sort of semantic structure to produce smooth and interpretable results.

F. Exploring the Disentanglement Property

Generating disentangled codes is a desirable property in the domain of representation learning in which specific properties in the input data are assigned to specific dimensions in the latent code during encoding. In the context of the proposed framework this would be achieved if the 3D shape properties such as height, length, and other spatial properties are separated/factorized in the compressed latent code. Practically this is encouraged during training through the additional factor β (only when β is above 1) in (2) as argued for in [30]. To explore the disentanglement of the learned latent space in the proposed method, a latent code is sampled, had all its dimensions fixed except for one dimension which is changed repeatedly, then finally the modified versions of the code are decoded to generate their corresponding meshes. Qualitative results of this process are shown in Fig. 6 which shows that the proposed method successfully produces disentangled codes that have some specific shape properties factorized.

V. CONCLUSION

This work proposes an improved method for the 3D generation of man-made 3D mesh shapes. This is achieved through using a Variational Autoencoder that encodes input meshes into a probabilistic latent space instead of a deterministic code. The decoder, on the other hand, can be fed a sampled code from the latent space and is capable of reconstructing novel and plausible variants of the input meshes. To evaluate the representational power of the proposed framework, several experiments were carried out including exploring the latent space to generate random samples, performing interpolation between two sampled points from the latent space, performing arithmetic operations on sampled meshes, and finally evaluating the disentanglement properties of the latent space. The experimental results show that the proposed framework is able to produce a smooth and traversable latent space that allows for all of these tasks. The main shortcoming of the proposed method is that it requires input meshes to be of the same topology and the same number of vertices. Overcoming this drawback is a possible venue for future work. Another possible venue for future work is working with meshes beyond genus-0 to cater for more variability in represented shapes.

REFERENCES

[1] W. Cao, Z. Yan, Z. He, and Z. He, "A comprehensive survey on geometric deep learning," IEEE Access, pp. 35929–35949, 2020.

[2] G. Fahim, K. Amin, and S. Zarif, "Single-View 3D reconstruction: a survey of deep learning methods," Computers & Graphics, vol. 94, pp. 164–190, Feb. 2021.

[3] A. X. Chang et al., "ShapeNet: an information-rich 3D model repository," arXiv:1512.03012 [cs], Dec. 2015.

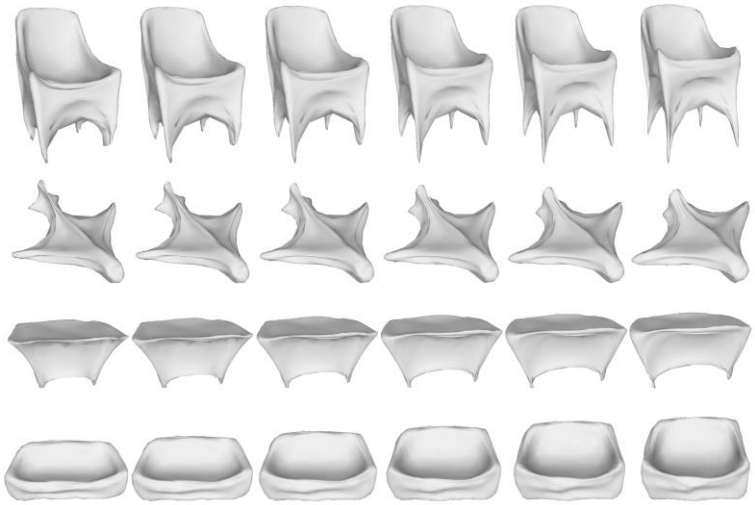

Fig. 6. Exploring the disentanglement of the latent representation. Each row consists of a sampled code that has all its dimensions fixed except for one dimension only. Changing this dimension changes a specific aspect in the decoded mesh. The chair appears to change the height of its legs, the plane gets shorter in length, the table's legs get more spread apart, the sofa increases its height.

[4] S. Koch et al., "ABC: a big CAD model dataset for geometric deep learning," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun. 2019, pp. 9593–9603.

[5] F. Bogo, J. Romero, M. Loper, and M. J. Black, "FAUST: dataset and evaluation for 3D mesh registration," in 2014 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun. 2014, pp. 3794–3801.

[6] S. Cheng, I. Kotsia, M. Pantic, and S. Zafeiriou, "4DFAB: a large scale 4D database for facial expression analysis and biometric applications," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun. 2018, pp. 5117–5126.

[7] C. Nash, Y. Ganin, S. M. A. Eslami, and P. Battaglia, "PolyGen: an autoregressive generative model of 3D meshes," in International Conference on Machine Learning, Nov. 2020, pp. 7220–7229.

[8] A. Sinha, A. Unmesh, Q. Huang, and K. Ramani, "SurfNet: generating 3D shape surfaces using deep residual networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition, Jul. 2017, pp. 791–800.

[9] D. Pavllo, G. Spinks, T. Hofmann, M. Moens, and A. Lucchi, "Convolutional generation of textured 3D meshes," in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020.

[10] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2Mesh: generating 3D mesh models from single RGB images," in Computer Vision – ECCV 2018, Cham, 2018, pp. 55–71.

[11] E. Smith, S. Fujimoto, A. Romero, and D. Meger, "GEOMetrics: exploiting geometric structure for graph-encoded objects," in Proceedings of the 36th International Conference on Machine Learning, 2019, vol. 97, pp. 5866–5876.

[12] S. Gong, L. Chen, M. Bronstein, and S. Zafeiriou, "SpiralNet++: a fast and highly efficient mesh convolution operator," in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Oct. 2019, pp. 4141–4148.

[13] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D faces using convolutional mesh autoencoders," in Computer Vision – ECCV 2018, Cham, 2018, pp. 725–741.

[14] Y.-J. Yuan, Y.-K. Lai, J. Yang, Q. Duan, H. Fu, and L. Gao, "Mesh variational autoencoders with edge contraction pooling," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jun. 2020, pp. 1105–1112.

[15] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," IEEE Signal Processing Magazine, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[16] D. P. Kingma and M. Welling, "Auto-Encoding variational bayes," arXiv:1312.6114 [cs, stat], May 2014.

[17] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative adversarial modeling," in Proceedings of the 30th International Conference on Neural Information Processing Systems, Dec. 2016, pp. 82–90.

[18] I. J. Goodfellow et al., "Generative adversarial nets," in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, Dec. 2014, pp. 2672–2680.

[19] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in Computer Vision – ECCV 2016, Cham, 2016, pp. 484–499.

[20] P. Mandikal, K. L. Navaneet, M. Agarwal, and R. V. Babu, "3D-LMNet: latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image," arXiv:1807.07796 [cs], Mar. 2019.

[21] P. Henderson and V. Ferrari, "Learning to generate and reconstruct 3D meshes with only 2D supervision," in British Machine Vision Conference 2018, BMVC 2018, September 3-6, 2018.

[22] G. Fahim, K. Amin, and S. Zarif, "Single-View 3D mesh reconstruction and generation," in Proceedings of the International Conference on Artificial Intelligence and Computer Vision, vol. 1377. Cham, 2021, pp. 473–482.

[23] Q. Tan, L. Gao, Y.-K. Lai, and S. Xia, "Variational autoencoders for deforming 3D mesh models," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun. 2018, pp. 5841–5850.

[24] I. Lim, A. Dielen, M. Campen, and L. Kobbelt, "A simple approach to intrinsic correspondence learning on unstructured 3D meshes," in Computer Vision – ECCV 2018 Workshops, Cham, 2019, pp. 349–362.

[25] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. CohenOr, "MeshCNN: a network with an edge," ACM Trans. Graph., vol. 38, no. 4, pp. 1–12, Jul. 2019.

[26] L. Gao et al., "SDM-NET: deep generative network for structured deformable mesh," ACM Trans. Graph., vol. 38, no. 6, pp. 1–15, Nov. 2019.

[27] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97, 1997, pp. 209–216.

[28] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv:1412.6980 [cs], Jan. 2017.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv:1301.3781 [cs], Sep. 2013.

[30] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "Beta-vae: learning basic visual concepts with a constrained variational framework," in 5th International Conference on Learning Representations, ICLR 2017, April 24-26, 2017.