Eurographics Workshop on 3D Object Retrieval (2022)
S. Berretti, T. Thehoaris, M. Daoudi, C. Ferrari, and R. C. Veltkamp (Editors)

Short Paper

Parameterization Robustness of 3D Auto-Encoders

E. Pierson[1†], T. Besnier[1,†], M.Daoudi[1,2] and S. Arguillère[3]

[1]Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
[2]IMT Nord Europe, Institut Mines-Télécom, Centre for Digital Systems
[3]Univ. Lille, CNRS, UMR 8524 Laboratoire Paul Painlevé, Lille, F-59000, France

Abstract

*The generation of 3-dimensional geometric objects in the most efficient way is a thriving research topic with, for example, the development of geometric deep learning, extending classical machine learning concepts to non euclidean data such as graphs or meshes. In this short paper, we study the effect of a reparameterization on two popular mesh and point cloud neural networks in an auto-encoder mode: PointNet [QSMG16] and SpiralNet [BBP*19]. Finally, we tested a modified version of PointNet that takes orientation into account (through coordinates of the normals) as a first step towards the construction of a geometric deep learning model built with a more flexible metric regarding the parameterization. The experimental results on standardized face datasets show that SpiralNet is more robust to the reparametrization than PointNet in this specific context with the proposed reparameterization.*

CCS Concepts

• ***Deep learning*** → *3D generative models;* • ***Performance measure*** → *Reparameterization; Robustness;*

1. Introduction

3D surfaces have known for a long time a large interest in the computer vision community. While the information contained in the 3D surface of a volumetric object is limited (because of the complexity of capturing and storing such data), the 3D mesh format has become one of the principal approaches. This is the case in human shape analysis, where data is generally presented as 3D scans of the human body or face resulting in a detailed representation of the shape. In particular, the shapes we consider are supposed to be smooth manifolds. This way, we reduce the complexity inherent to volumetric data, though we do lose some information. *Point clouds* or *meshes* sampled from the original shape surfaces are natural discretizations for describing the corresponding continuous objects.

In recent years, the computer vision community has become increasingly interested in deep learning approaches, with the success of convolutional neural networks on 2D images [KSH12, SLJ*15, HZRS16], and models that are now well established, such as auto-encoders. In the meantime, the quality and quantity of available 3D data has exploded, unveiling the possibility of training and applying popular neural networks on 3D data, with multiple way of representing and learning 3D shapes [BLRW16, GMW17, FSG17, ADMG17, KTEM18, RBSB18].

Moreover, modeling complex 3D object such as human body and faces is even more challenging as human data undergoes specific and complex transformations. For example, human faces are usually described as the sum of identity and facial expression as smooth deformations from a neutral template face. Two different faces can be linked from one to another through the combinations of smooth deformations. In the general setting, modeling those deformations can be done using costly procedures such as diffeomorphic registration or linear blending models. The possibility to encode such deformations or generate shapes (faces, bodies, ...) resembling the ones from a given database with a neural network in a *auto-encoder* architecture sounds like the next pathway. In order to fullfill this goal, the possibility of applying CNNs directly on the surface [BBL*17], instead of voxels, is promising. However from this problem arises several challenges due in part to natural invariances present in 3D objects.

Among the challenges raised by 3D learning on explicit representations, dealing with reparameterization is crucial for real world applications. In this paper, we will refer to the *reparameterization* of a mesh as the re-sampling of its corresponding surface. Compared to images, is not easy to find a canonical correspondence between two meshes (generally, a scan and a template), and powerful registration algorithms are often needed to build in correspondence mesh datasets. While full 3D auto encoding of unregistered meshes remain an active area of research, several architectures have shown state-of-the-art results for non lossy encoding of 3D meshes. Recent work [OFD*22] suggests however that some 3D auto-encoders struggle when the meshes presented uncanny deformations: hard poses for 3D human poses, or new identities for 3D faces.

† Equal Contribution

DOI: 10.2312/3dor.20221180

Interested by the problem of parameterization we propose a benchmark on the COMA dataset [RBSB18] to compare the robustness against such changes, in the registered setting, on two auto-encoders.

2. Related work

2.1. Deformation of 3D meshes

Being able to represent faithfully a 3D shape is a hard task. For a long time, handcrafting descriptors has been a preferred approach, with successful approaches in representing global shape properties such as the Heat Kernel Signature [BK10] or local descriptors such as the shape index [KVD92, CLZY15]. The way we design an auto encoder of 3D meshes in a registered setting, is as a deformation of a template mesh. This area has been active for several years. In particular, several linear blending models have shown the ability to be able to represent human shapes while having a good generalisation ability [ASK*05, LBB*17]. However, the retrieval of a parameter for a single shape is a costly procedure, that often needs human intervention or additional data (landmarks, texture videos, ...).

2.2. Geometric deep learning

In the meantime, the field of geometric deep learning [BBL*17] has become very important. The purpose of this area is to design new feature extractors from geometrical data such as graphs, and 3D shapes in supposedly all kind of formats (point clouds, meshes). After the promising results of PointNet [QSMG16], local shape descriptors such as PointNet++ [QYSG17] or KPConv [TQD*19] has shown impressive results on point clouds for shape classification or scene segmentation. While this area remains very active, we refer to [GWH*20] for a broad review of developments in this area. Meanwhile, applying the principle of 2D image convolutions on 3D surface meshes is today a hot topic. Pioneer work such as GCNN or FeastNet showed impressive results for mesh classification or local feature extraction. Most of these approaches however rely on k-NN (for point clouds), n-ring (for meshes) feature aggregators, or careful remeshing in training in order to work properly. Moreover, they are often too heavy for the training of deep neural auto-encoders. The recent DiffusionNet approach [SACO22], which relies on efficient approximation of surface intrinsic diffusion using the discrete cotangent Laplacian, is promising.

2.3. Mesh auto-encoders

The geometric deep feature extractors opens the possibility of learning representations of shapes directly from the data using, for example, unsupervised auto encoders [BBP*19, HHS*21, CNH*20, ZWL*20, HHF*19]. Nowadays, Neural3DMM [BBP*19, CK21] or SpiralNet, based on the idea of Spiral Convolution, is a popular approach. However, this approach is limited by its need to work on registered shapes. On the other side, the well established PointNet [QSMG16] architecture shows good generalisation abilities over point cloud analysis, and is applied on a wide variety of problems, including auto-encoders [ADMG18, CNH*20]. PointNet stays still limited by the fact that it completely ignores the connectivity of a mesh, and thus the intrinsic properties of the surface it encodes. Finally, recent approaches of implicit network for representing surfaces is an active area, in particular for shape reconstruction from images [SHN*19], or partial 3D scan [CAPM20]. However, the encoding of 3D shapes mostly rely on heavy 3D CNN operators on voxels [CAPM20], thus they fail to capture information on detailed shapes such as 3D faces.

2.4. Contribution

The purpose of this work is to test the robustness of two auto-encoders with respect to a specific reparameterization. To do so, we will train two popular auto encoders (SpiralNet and PointNet) on the COMA dataset, Then, we test the trained network on both COMA and FLAME [LBB*17] datasets, which is a reparametrization of the surfaces in COMA. We expected PointNet to be much worse than SpiralNet on the original dataset, but that SpiralNet would break down on FLAME, since it is so dependent on mesh connectivity. However, this was not the case on the proposed reparameterization for the face generation task. Our conclusions are summarized as follow:

1. As expected, the PointNet architecture is worse at encoding data compared to the SpiralNet registered approach on the original parameterization
2. The SpiralNet model breaks when the reparameterization changes the graph structure on our artificial database.
3. The SpiralNet architecture is surprisingly robust on the proposed reparameterization for face generation.

3. Auto Encoders and their characteristic

Consider a discretized surface M of a 3D face shape (point cloud or mesh), a latent space Z and the output surface N. The goal of an auto-encoder is to learn an encoder $\Phi : M \rightarrow Z$, and a decoder $\psi : Z \rightarrow N$. In addition, to make use of all available data, an ideal auto-encoder should be invariant with respect to several transformations, which we highlight below. Those invariances are among the biggest current challenges in geometric deep learning.

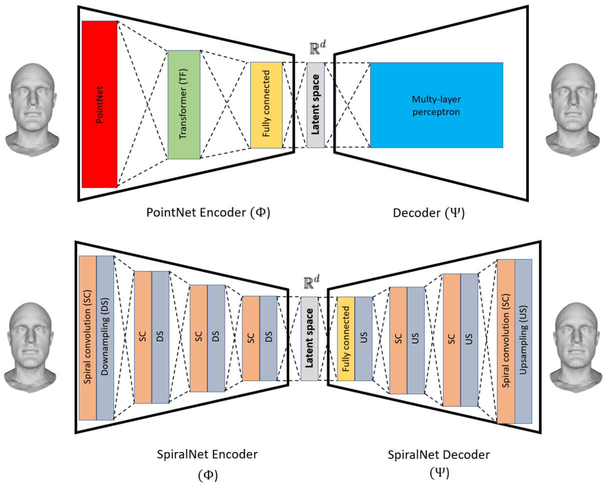

Figure 1: *SpiralNet and PointNet Encoders/Decoders*

3.1. Invariance to Euclidean Tranformations

For a geometric model, we often seek invariance to isometries such as *rotations* and *translations* in $\mathbb{R}^3$. We denote $\mathcal{T}$ the group of translations and $SO(3)$ the group of rotations. We aim at having, for each $T \in \mathcal{T}, R \in SO(3)$

$$\Psi \circ \Phi(T.M) = \Psi \circ \Phi(M) \tag{1}$$

$$\Psi \circ \Phi(R.M) = \Psi \circ \Phi(M) \tag{2}$$

The design of neural networks invariant to such transformation is a very active area of research with a lot of progress in the recent years. We restrict ourselves to already aligned shapes (centered, canonically rotated) and left this problem out of the scope of this paper.

3.2. Invariance to Parametrization

Permutations. When working only with point cloud, the parameterization can be seen as a specific ordering of a point cloud. The invariance is such that the reordering of each sample point should not impact the performance of the model. More precisely, if we consider $x = \{x_1, ..., x_n\}$ a set of sample points describing M. Let $\mathcal{P}$ be the group of permutations, for $\sigma \in \mathcal{P}$ such that $\sigma(x) = \{x_{\sigma(1)}, ..., x_{\sigma(n)}\}$

$$\Psi \circ \Phi(\sigma(x)) = \Psi \circ \Phi(x) \tag{3}$$

In this setting, the invariance is successfully solved by PointNet as shown in [QSMG16].

General setting. In the general setting, we rather talk about the invariance of a function applied to different discretizations. Given M_1 and M_2, two different discretizations representing the same 3D surface, we would like

$$\psi \circ \phi(M_1) = \psi \circ \phi(M_2)$$

One should keep in mind that this is an ideal condition and so, we are interested in a robust model with the following property

$$|\psi \circ \phi(M_1) - \psi \circ \phi(M_2)| \quad \text{is bounded} \tag{4}$$

where $|\cdot|$ is a given metric between discretized shapes.
Testing this robustness is the goal of this paper. We will however remain in the simplified setting of same topology between two meshes. A reparameterization is then seen as a translation of the mesh over the continuous surface.

4. Neural Networks for Point Cloud and Meshes

4.1. PointNet

PointNet [QSMG16] is an architecture dealing with point clouds that learns a symmetric function of the point cloud capturing properties with the constraints mentioned before and summarising the point cloud to a fixed size vector by using a max pooling operation. A summary illustration is proposed in Figure 2

PointNet has two advantages: its simplicity (we only apply Multi-layer perceptrons), which makes training and inference computationally efficient, and its natural reparameterization-invariance. [QSMG16]

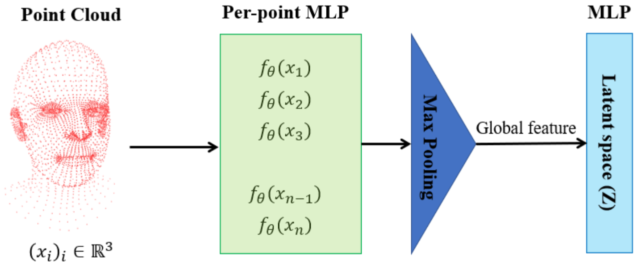

Figure 2: *Architecture of the PointNet Encoder. θ and λ are the learned variables of the MLPs*

Our auto-encoder is the simplified architecture proposed in [ADMG18], which has been applied successfully to 3D faces in [CNH*20]. The decoder part is simply a Multi-layer Perceptron (MLP) that reconstructs the points directly from the latent space vector.

4.2. Modified PointNet model

We enhanced the PointNet architecture with normals coordinates as additional dimensions of the input vector, similar to [QYSG17]. We use the normals attached to each vertices. This way, we use the information contained in the triangulation of the mesh instead of the vanilla point cloud.

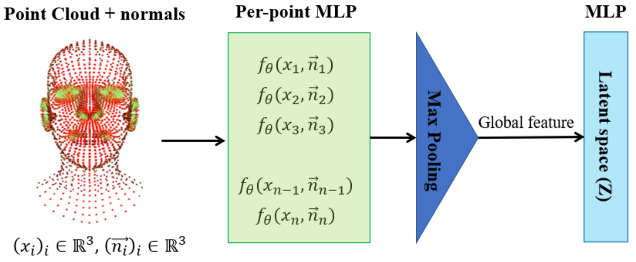
Point Cloud + normals
Per-point MLP
MLP
$f_\theta(x_1, \vec{n}_1)$
$f_\theta(x_2, \vec{n}_2)$
$f_\theta(x_3, \vec{n}_3)$
$f_\theta(x_{n-1}, \vec{n}_{n-1})$
$f_\theta(x_n, \vec{n}_n)$
Max Pooling
Global feature
Latent space (Z)
$(x_i)_i \in \mathbb{R}^3$, $(\vec{n_i})_i \in \mathbb{R}^3$

Figure 3: *Architecture of our modified version of PointNet with normals*

The decoder architecture is similar to the standard PointNet auto-encoder described earlier. We illustrate it in Figure 3.

4.3. Spiral convolution and mesh pooling

We define the face mesh as in [BBP*19] as a graph $\mathcal{M} = (\mathcal{V}, \mathcal{E})$; where $\mathcal{V} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and $\mathcal{E}$ respectively denote the set of edges and vertices.

The model described in [BBP*19] is a model dealing with geometric data described as graphs. This model propose a convolution layer that deals with ordering problems. To do so, the mesh is described from a template on which is attached a *spiral patch*. This patch impose a local ordering respected by the usual graph convolution operations. The figure 4 shows two examples of spiral patch around two given vertices (in red).

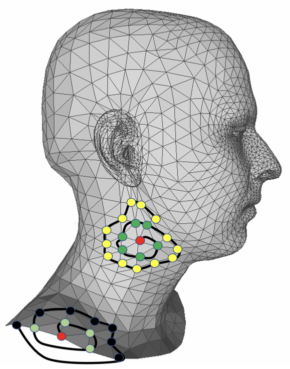

Figure 4: *An example of two spiral neighborhoods around a vertex on the facial mesh*

More formally, from a given point $x \in \mathcal{V}$, a spiral patch is an operator S

$$S(x) = \{x, R_1^1(x), R_2^1(x), \dots, R_{|R^h|}^h(x)\}$$

where $R_j^i(x)$ is the j-th element of the i-ring around x. The patch starts from x, $R_1^1(x)$ is the closest point from x on the 1-ring, then the ring is visited with a fixed travel direction (clockwise or counter-clockwise), the process is repeated until the h-ring with h an hyper-parameter of the model.

This operator is then used for the convolution operation which becomes, for two signals f, g on the graph $\mathcal{M}$

$$(f * g)_x = \sum_{l=1}^{L} g_l f(S_l(x))$$

Following the convolution layer, the mesh pooling operation, described in [RBSB18] is a downsampling of the mesh obtained by removing vertices that minimize a quadric error inspired by [GH97].

In particular, robustness against reparameterization is not guaranteed (and expected) for this model.

5. Numerical experiments

With an auto-encoder version of each model, described below and summarised in Figure 1 we aim at reproducing 3D faces with fixed number of vertices.

5.1. Datasets

ELLIPSES is an artificial database made of three different parameterizations of 1000 randomly generated ellipsoids. We start by taking a random parameterization of the sphere (we call it the original parameterization) and we apply a random diagonal matrix to get an ellipse. We repeat this process 1000 time to build the dataset and we store each transformation. To create a reparameterization we rotate the original sphere and apply the stored transformation matrices to get a "shifted" reparameterization. Finally, to get an "unbalanced" reparameterization, we unevenly sample on the sphere (for example, five times more sample on one side) and we apply the stored transformations to obtain the unbalanced reparameterization.

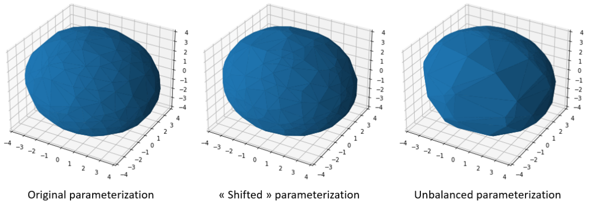
Original parameterization
« Shifted » parameterization
Unbalanced parameterization

Figure 5: *A sample from each dataset in the ELLIPSES database*

COMA dataset [RBSB18] is a dataset of 3D faces. 12 persons executed at most 12 different "extreme" expressions (*mouth extreme, mouth middle, mouth open, mouth side, mouth up, lips up, lips back, high smile, eyebrow, cheeks in, bare teeth*) displayed in Figure 6. The videos of each expression are scanned using a 4D scanner and a registration method is applied to get registered meshes in the FLAME topology (see below). It is composed in total of 20466 meshes, obtained from 3D scans, of 12 different persons (identities). The sequences are sampled at 144HZ, and each expression sequence contains between 29 and 150 meshes.

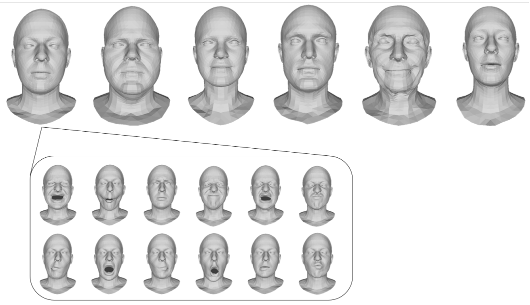

Figure 6: *Examples of meshes in the COMA dataset*

FLAME [LBB*17]is a framework to model 3D faces through 3 types of transformations: shape (or identity), pose and expression. A template mesh (5063 vertices) is deformed along each transformation direction to reconstruct a face. The FLAME model is learned over more that 30000 real 3D face scans from the D3DFACS [CKH11] database. We applied the FLAME pipeline on the registered meshes from the COMA dataset, resulting in a slightly different parameterization of the faces (in the FLAME directions), see Figure 7 . The topology is however the same in both COMA and FLAME datasets.

5.2. Experimental setup

Each of the following auto-encoders is trained for 250 epochs with a latent space of fixed size 128. The loss used both for training and

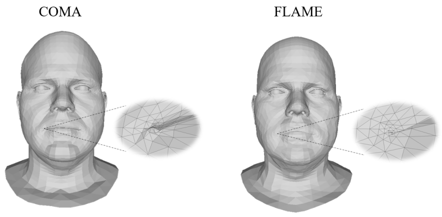

Figure 7: *Reparameterization of one COMA registered face (left) using the FLAME model (right). We observe that the obtain vertex are slightly translated on the surface compared to the original COMA registrations.*

evaluation is $d^1(x,\hat{x}) = \frac{1}{N}\sum_i |x_i - \hat{x}_i|$. Optimization is performed with a standard Adam optimizer and a learning rate of 10^{-3}.

For the ellipsoid generation, the model is trained on 90% of the "original" parameterization and tested on the remaining 10%. Then the trained model is tested on the whole "shifted" and "unbalanced" datasets.

For the face generation task, we used 11 out of the 12 identities of the COMA dataset as training set and the remaining identity is used for the test phase which gives the numbers in the "COMA" column in **??**. The performance is then compared by applying the trained model to the corresponding identity of the FLAME dataset, which gives us the "FLAME" column in **??**. We describe the parameters of the model for the face generation task. For the ellipsoid generation task, the model has half the number of parameters but keeps the same structure.

SpiralNet Auto-encoder: Filters size for the encoder are [64,64,64,128] with a downsampling (DS) of factor 4 after each spiral convolution (SC). The decoder is a mirror of the encoder starting with a fully connected layer (FC). The parameters are taken from the original Neural3DMM [BBP*19] paper.

- *Enc:* SC(64) → DS(4) → SC(64) → DS(4) → SC(64) → DS(4) → SC(128) → DS(4)
- *Dec:* FC(128) → US(4) → SC(128) → US(4) → SC(64) → US(4) → SC(64) → US(4) → SC(64)

PointNet Auto-encoder: The encoder is a combination of a simple PointNet architecture (PN), a transformer (TF) and a fully connected layer (FC). The decoder is a simple Multi-layer perceptron (MLP). We use the parameters of [CNH*20], that were optimized on the COMA dataset.

- *Enc:* PN(64,128) → TF(64) → FC(128,64,64)
- *Dec:* MLP(256)

5.3. Quantitative experiments

We run experiments by training our encoders on both ELLIPSES (original parameterization) and COMA datasets. We summarise our results in **??** for tests on the ELLIPSES database and in Table 2 for tests on COMA and FLAME.

Ellipse generation: We aim at generating ellipsoids with both auto-encoders after a training phase on the original parameterization from which is derived the template used for the SpiralNet model. The performances regarding robustness against the different reparameterizations are evaluated in Table 1.

	Original	Shifted	Unbalanced
SpiralNet	0.01	102.02	149.22
PointNet	0.80	1.41	228.48

Table 1: *Results regarding robustness over reparameterization tested on the ELLIPSES database*

We observe that SpiralNet breaks as soon as the graph structure changes compared to the template (on shifted and unbalanced parameterizations). In comparison, PointNet maintains its ability to reconstruct an ellipse for a shifted reparameterization but the model breaks for the unbalanced parameterization (see Figure 8).

Face generation: Similarly, to generate faces, with the setup described before, we trained the two models on a given parameterization (COMA) and we evaluated the performance on both COMA and FLAME parameterizations. The results are presented in Table 2.

	COMA	FLAME
SpiralNet		
1st identity	0.350	0.381
2nd identity	0.293	0.305
3rd identity	0.334	0.365
PointNet		
1st identity	0.703	0.659
2nd identity	0.653	0.680
3rd identity	0.819	0.817
Modified PointNet		
1st identity	0.644	0.738
2nd identity	0.685	0.772
3rd identity	0.720	0.812

Table 2: *Results regarding robustness over reparameterization tested on 1 over 12 identity*

We observe in this table that the both PointNet et SpiralNet auto-encoders are not affected by FLAME reparameterization.

For the modified PointNet model, although we add more information into the input of the network, this experiment shows that the model using normals has better performance with data from the same parameterization but fails to improve its performance on reparameterized data. This is probably due to the fact that the normals coordinates can be seen as information linked to the differential of the surface which is more sensible to small deformations.

5.4. Qualitative results

Finally, we display some reconstructions of the meshes in the test set in Figure 9.

Regarding performances in the ellipsoid reconstruction task, we highlight in Figure 8 an example summarising how the model performs under each type of transformation. For face reconstruction,

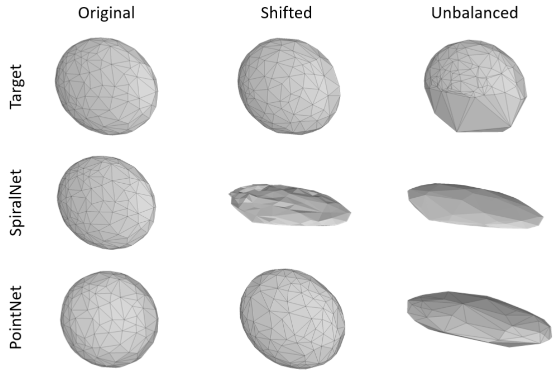

Figure 8: *Examples of different parameterizations of an ellipse*

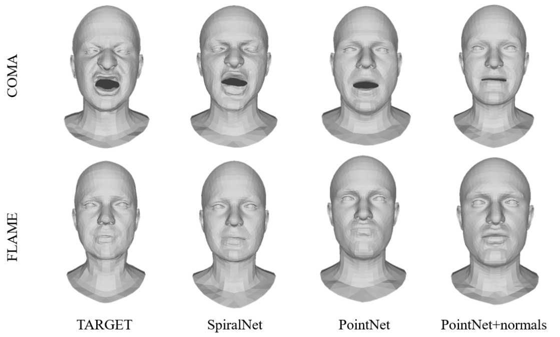

Figure 9: *Examples of reconstruction for each model*

we observe that SpiralNet is the only one keeping the original encoded identity in both settings. Moreover, in the COMA setting, the PointNet model is able to keep a part of the expression, while loosing the identity, while the augmented model seems to behave differently (the expression is totally different while the identity is a bit similar). In the FLAME setting, both models fails completely, in expression and identity.

6. Conclusion and future work

In the present work, we discuss the problem of reparameterization for auto-encoders. We experimented the effect of a specific reparameterization for two different models. First, experiments conducted on a simple artificial database show that, in general, SpiralNet should not be expected to be robust against reparameterization. The experimental results on COMA (used for training and test) and FLAME (only used for test) datasets show that, regardless of our expectations, SpiralNet is robust against the aforementioned reparametrization which has the specificity to maintain a similar graph structure. In addition, PointNet which is supposedly reparameterization-invariant for classification and segmentation tasks, is robust but not completely invariant when used as encoder in the context of our experiments. This paper is a first step for the larger goal consisting in the generation of 3D faces using a deep learning architecture that is robust against a reparameterization of the mesh.

In future work, we seek to extend this benchmark to a broader class of mesh auto encoders, and to build a benchmark for the fully unparameterized meshes setting. In this perspective, the exploration of tools such as geometric measure theory [Gla05,KCC17] in order to define parameterization robust metrics is a promising path.

References

[ADMG17] Achlioptas P., Diamanti O., Mitliagkas I., Guibas L.: Learning representations and generative models for 3d point clouds. URL: https://arxiv.org/abs/1707.02392, doi:10.48550/ARXIV.1707.02392. 1

[ADMG18] Achlioptas P., Diamanti O., Mitliagkas I., Guibas L.: Learning representations and generative models for 3d point clouds. In *International conference on machine learning* (2018), PMLR, pp. 40–49. 2, 3

[ASK*05] Anguelov D., Srinivasan P., Koller D., Thrun S., Rodgers J., Davis J.: Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*. 2005, pp. 408–416. 2

[BBL*17] Bronstein M. M., Bruna J., LeCun Y., Szlam A., Vandergheynst P.: Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine 34*, 4 (jul 2017), 18–42. doi:10.1109/msp.2017.2693418. 1, 2

[BBP*19] Bouritsas G., Bokhnyak S., Ploumpis S., Bronstein M., Zafeiriou S.: Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *The IEEE International Conference on Computer Vision (ICCV)* (2019). 1, 2, 3, 5

[BK10] Bronstein M. M., Kokkinos I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition* (2010), IEEE, pp. 1704–1711. 2

[BLRW16] Brock A., Lim T., Ritchie J., Weston N.: Generative and discriminative voxel modeling with convolutional neural networks. 1

[CAPM20] Chibane J., Alldieck T., Pons-Moll G.: Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 6970–6981. 2

[CK21] Chen Z., Kim T.-K.: Learning feature aggregation for deep 3d morphable models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 13164–13173. 2

[CKH11] Cosker D., Krumhuber E., Hilton A.: A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. In *2011 International Conference on Computer Vision* (2011), pp. 2296–2303. doi:10.1109/ICCV.2011.6126510. 4

[CLZY15] Canavan S., Liu P., Zhang X., Yin L.: Landmark localization on 3d/4d range data using a shape index-based statistical shape model with global and local constraints. *Computer Vision and Image Understanding 139* (2015), 136–148. 2

[CNH*20] Cosmo L., Norelli A., Halimi O., Kimmel R., Rodola E.: Limp: Learning latent shape representations with metric preservation priors. In *European Conference on Computer Vision* (2020), Springer, pp. 19–35. 2, 3, 5

[FSG17] Fan H., Su H., Guibas L.: A point set generation network for 3d object reconstruction from a single image. pp. 2463–2471. doi:10.1109/CVPR.2017.264. 1

[GH97] Garland M., Heckbert P.: Surface simplification using quadric error metrics. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics 1997* (07 1997). doi:10.1145/258734.258849. 4

[Gla05] Glaunès J. A.: *Transport par difféomorphismes de points, de mesures et de courants pour la comparaison de formes et l'anatomie* *numérique*. PhD thesis, Université Sorbonne Paris-Nord, 2005. Thèse de doctorat dirigée par Younes, Laurent et Trouvé, Alain Mathématiques Paris 13 2005. URL: http://www.theses.fr/2005PA132033. 6

[GMW17] GADELHA M., MAJI S., WANG R.: 3d shape induction from 2d views of multiple objects. In *2017 International Conference on 3D Vision (3DV)* (Los Alamitos, CA, USA, oct 2017), IEEE Computer Society, pp. 402–411. URL: https://doi.ieeecomputersociety.org/10.1109/3DV.2017.00053, doi:10.1109/3DV.2017.00053. 1

[GWH*20] GUO Y., WANG H., HU Q., LIU H., LIU L., BENNAMOUN M.: Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence 43*, 12 (2020), 4338–4364. 2

[HHF*19] HANOCKA R., HERTZ A., FISH N., GIRYES R., FLEISHMAN S., COHEN-OR D.: Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–12. 2

[HHS*21] HUANG Q., HUANG X., SUN B., ZHANG Z., JIANG J., BAJAJ C.: Arapreg: An as-rigid-as possible regularization loss for learning deformable shape generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5815–5825. 2

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA, USA, jun 2016), IEEE Computer Society, pp. 770–778. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.90, doi:10.1109/CVPR.2016.90. 1

[KCC17] KALTENMARK I., CHARLIER B., CHARON N.: A General Framework for Curve and Surface Comparison and Registration with Oriented Varifolds. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, United States, July 2017), IEEE. URL: https://hal.archives-ouvertes.fr/hal-01817514, doi:10.1109/CVPR.2017.487. 6

[KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (2012), Pereira F., Burges C., Bottou L., Weinberger K., (Eds.), vol. 25, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf. 1

[KTEM18] KANAZAWA A., TULSIANI S., EFROS A. A., MALIK J.: Learning category-specific mesh reconstruction from image collections. In *ECCV* (2018). 1

[KVD92] KOENDERINK J. J., VAN DOORN A. J.: Surface shape and curvature scales. *Image and vision computing 10*, 8 (1992), 557–564. 2

[LBB*17] LI T., BOLKART T., BLACK M. J., LI H., ROMERO J.: Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia) 36*, 6 (2017), 194:1–194:17. URL: https://doi.org/10.1145/3130800.3130813. 2, 4

[OFD*22] OTBERDOUT N., FERRARI C., DAOUDI M., BERRETTI S., DEL BIMBO A.: Sparse to dense dynamic 3d facial expression generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), pp. 20385–20394. 1

[QSMG16] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593* (2016). 1, 2, 3

[QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems 30* (2017). 2, 3

[RBSB18] RANJAN A., BOLKART T., SANYAL S., BLACK M. J.: Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)* (2018), pp. 725–741. URL: http://coma.is.tue.mpg.de/. 1, 2, 4

[SACO22] SHARP N., ATTAIKI S., CRANE K., OVSJANIKOV M.: Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG) 41*, 3 (2022), 1–16. 2

[SHN*19] SAITO S., HUANG Z., NATSUME R., MORISHIMA S., KANAZAWA A., LI H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 2304–2314. 2

[SLJ*15] SZEGEDY C., LIU W., JIA Y., SERMANET P., REED S., ANGUELOV D., ERHAN D., VANHOUCKE V., RABINOVICH A.: Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 1–9. doi:10.1109/CVPR.2015.7298594. 1

[TQD*19] THOMAS H., QI C. R., DESCHAUD J.-E., MARCOTEGUI B., GOULETTE F., GUIBAS L. J.: Kpconv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE International Conference on Computer Vision* (2019). 2

[ZWL*20] ZHOU Y., WU C., LI Z., CAO C., YE Y., SARAGIH J., LI H., SHEIKH Y.: Fully convolutional mesh autoencoder using efficient spatially varying kernels. *Advances in Neural Information Processing Systems 33* (2020), 9251–9262. 2