On Variational Auto-Encoders for Fixed Graph Mesh Learning

Nicolas Vercheval[1,2], Hendrik De Bie[2], and Aleksandra Pižurica[1].

[1]Department of Telecommunications and Information Processing, TELIN-GAIM, Faculty of Engineering and Architecture, Ghent University, Belgium

[2] Department of Electronics and Information Systems, Clifford Research Group, Faculty of Engineering and Architecture, Ghent University, Belgium.

***Abstract*— In this paper, we propose a Variational Auto-Encoder able to correctly reconstruct a fine mesh from a very low-dimensional latent space. The architecture avoids the usual coarsening of the graph and relies on pooling layers for the decoding phase and on the mean values of the training set for the up-sampling phase. We select new operators compared to previous work, and in particular, we define a new Dirac operator which can be extended to different types of graph structured data. We show the improvements over the previous operators and compare the results with the current benchmark on the Coma Dataset.**

1 Introduction

Meshes provide a compact graphical 3D-representation of a surface, and Geometric Deep Learning [1] seizes the additional graphical information to extend the traditional deep learning framework to non-Euclidean domains. This is of interest both for supervised tasks, typically classification, and unsupervised tasks such in generative modelling.

Recently, Variational Auto-Encoders (VAE) [5] have been successfully combined with Chebnets [3] to approximate the distribution of the Coma Dataset[7], assumed to be governed only by 8 latent variables, and in producing realistic new meshes of human facial expressions. To create the VAE bottleneck they use progressive down-sampling and up-sampling of the nodes, which is a standard way to leverage the structure of the graph.

In Surface Networks [6], Kostrikov et al. only rely on differential operators to enforce the local structure of the graphs and combine the graph features using 1×1 Convolutions. This type of architecture is as versatile as a multilayer perceptron, and is grouped in residual blocks [4] to allow for deeper networks.

To reduce the dimensionality and create the latent variables, Surface Networks use global pooling on the graph features which are then symmetrically recreated by expanding the latent variables onto the graph. While adequate on a toy dataset, this architecture struggles to reproduce finer meshes with a larger amount of nodes. Furthermore, storing data dependent operators poses huge memory requirements.

In this article we exploit the fixed graph structure of a dataset by initiating the decoder with the mean values of the training set. The initialised decoder is able to generate fine smooth meshes which were unachievable with the previous architecture from [6], and we demonstrate its effectiveness by accurately encoding the Coma dataset.

Additionally, we replace the original cotangent Laplacian with an adjacent version, and introduce a new version of the Dirac operator, which squares to the normalized Laplacian and can be used for any chordal graph. As the adjacency matrix is shared, we can then use the same operator for all the samples. This not only relieves the memory burden but prevents the network from overfitting.

2 Operators

A mesh is a data structure comprising the embedding of the nodes $\phi\colon \mathbb{V} \to \mathbb{R}^3$, and a list of triangular faces $\mathbb{F}$. Given a node v, its dual area $a(v)$ is a third of the area A of the surrounding triangles.

Let W be the symmetric matrix of cotangent weights and a be the diagonal matrix of the dual areas. The degree matrix D is a diagonal matrix such that $D_{i,i} = \sum_j W_{i,j}$. The normalized graph Laplacian with cotangent weights, commonly used in mesh processing, is defined as: $\Delta = a^{-1}(D - W)$.

The symmetric Laplacian is defined instead as $\tilde{\Delta} = 1 - \tilde{L} = 1 - D^{1/2}\tilde{W}D^{1/2}$, where $\tilde{W}$ is the adjacency matrix and $\tilde{L}$ will be referred as the adjacency Laplacian.

The Dirac operator introduced by [2] is a discrete differential operator on quaternion-valued functions (the embedding is immersed in their imaginary part) between the graph and the dual graph. The dual graph is constructed from the faces of the original graph, where two connected (dual) nodes correspond to two adjacent faces.

On an oriented triangular mesh, the Dirac operator is defined on a quaternionic function $\lambda\colon \mathbb{V} \to \mathbb{H}$ as follows:

$$(Di_\phi\lambda)_F = -\frac{e_{(1,0)} \cdot \lambda(v_2) + e_{(2,1)} \cdot \lambda(v_0) + e_{(0,2)} \cdot \lambda(v_1)}{2A(F)},$$

where $e_{(i,j)} = \phi(v_i) - \phi(v_j)$ is an edge of $F = (v_0, v_1, v_2) \in \mathbb{F}$ and $\cdot$ is the quaternionic multiplication.

In the neural network it is used in combination with its adjoint operator $Di^A = a^{-1}Di^t A$ which maps the values on the dual nodes back to the original graph.

It can be shown that $\Re(Di_\phi^A Di_\phi) = \Delta$.

The symmetric Laplacian is defined as $\tilde{\Delta} = 1 - \tilde{L} = 1 - D^{1/2}\tilde{W}D^{1/2}$, where $\tilde{W}$ is the adjacency matrix and $\tilde{L}$ will be referred as the adjacency Laplacian.

As an alternative to the Dirac operator, we introduce the adjacent Dirac operator and we define it only through the adjacency matrix as $\widetilde{Di} = 2D^{1/2}Di_{\tilde{\phi}}$ where $\tilde{\phi}$ maps each cycle (triangle) to $\{v_0, v_1, v_2\}$ and where

$$v_0 = (\frac{1}{\sqrt{2}}, 0, 0),\ v_1 = (0, \frac{1}{\sqrt{2}}, 0),\ v_2 = (0, 0, \frac{1}{\sqrt{2}}).$$

Like the extrinsic version, defining $\widetilde{Di}^A = \widetilde{Di}^T$ it can be shown that

$$\Re(\widetilde{Di}^A \widetilde{Di}) = \tilde{\Delta}.$$

This operator can also be applied to more general chordal graphs.

3 Architecture

The Variational Auto-Encoders [5] are composed of an encoder and of a decoder, both approximated with neural networks. The

Decoder takes a noisy version of the encoded samples during training which makes it more robust while respecting the probabilistic interpretation of the model (the reparameterization trick).

The networks are composed of ResNet blocks with "1×1 convolution" layers, concatenation and multiplication with the operators, which in the encoder are then followed by max-pooling and dense layers to get the means and variances of the codes.

The variational loss is the opposite of the ELBO [5]:

$$\mathcal{L} = -\text{ELBO} = \underbrace{-\log p_\theta(x|z)}_{\text{reconstruction loss}} \underbrace{-\log p_\theta(z) + \log q_\phi(z|x)}_{\text{regularization terms}}.$$

3.1 Decoder

We observe that Surface Networks fail to converge on high-resolution meshes and are attracted to the barycenter of the sample. As the graph structure is only communicated through the use of the operators as matrix multiplication we infer that the residual blocks struggle to enforce information from the adjacent nodes.

To overcome this, we propose to initialize the decoder with a smooth mesh so that smoothness could easily be preserved. To take full advantage of the fixed structure we decide to use the mean shape, the mean embedding over all the nodes of the training set.

The mean shape and the latent variables are given to the decoder as a multimodal input. The proposed architecture joins the two initial inputs with a tensor multiplication as in Fig. 1. When decoding with the original operators we use the mean operator which is similarly calculated.

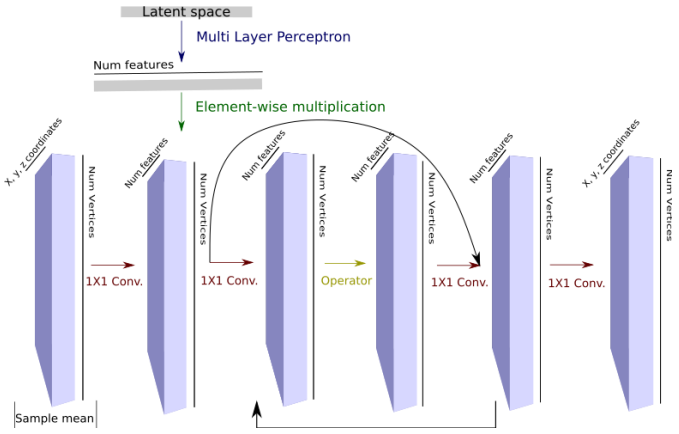

Figure 1: Proposed architecture of the Decoder. The 1×1 Convolution can be seen as a dense layer on the feature graphs and is preceded by Batch Normalization. The arrow in the bottom represents a Resnet block.

4 Experiments and Conclusion

The Coma Dataset [7] is composed of temporal sequences of extreme facial expressions by 12 subjects, recorded as meshes on a common graph.

Table 1 shows that the adjacent versions of the operators offer a clear improvement over the ones previously used. In particular the adjacent Laplacian approaches the performance of the method of [7], while having less weights. It also compares favourably to all the other approaches listed in a recent paper [8].

Since the original operators are calculated from the samples, they are able to express more information, but this also leads to instability in some areas (Fig.2). One explanation is that the nets are harder to optimize when their layers use operators who depend on the samples themselves.

When using the adjacent operators, the latent space is robust and able to represent the semantic information of the meshes. We show that by sampling the latent space (Fig. 3). A more extensive experimental evaluation is currently under submission in [9].

Table 1: Euclidean error and percentage of correct (<1 mm) node reconstructions by operator. The adjacent operators outperformed the ones used previously and approaches the benchmark. The vanilla Surface Networks did not converge.

Networks	***Operator***	***Error (mm)***	***% correct***	***# Weights***
Proposed	Di	1.90 ± 0.15	21.9	28,660
	$\widetilde{Di}$	1.47 ± 0.10	42.3	28,660
	Δ	2.08 ± 0.36	21.5	28,660
	$\tilde{L}$	1.10 ± 0.05	57.7	28,660
DEMEA[a]	$\tilde{\Delta}$	1.49	/	/
Coma[b]	$\tilde{\Delta}$	0.845 ± 0.99	72.6	33,856

[a] As reported in [8] (no confidence interval was provided).
[b] As reported in [7].

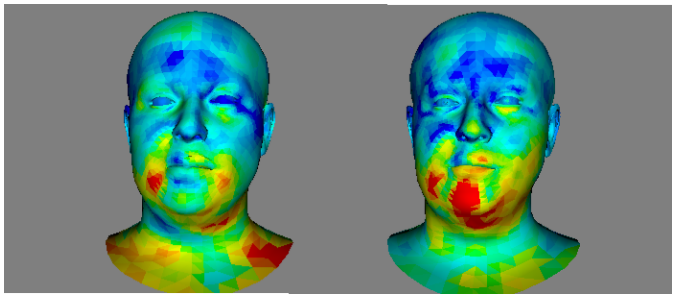

Figure 2: The same mesh reconstructed by the Δ and $\tilde{L}$. The warmer colors highlight the larger errors. Using Δ, those errors are often around the neck.

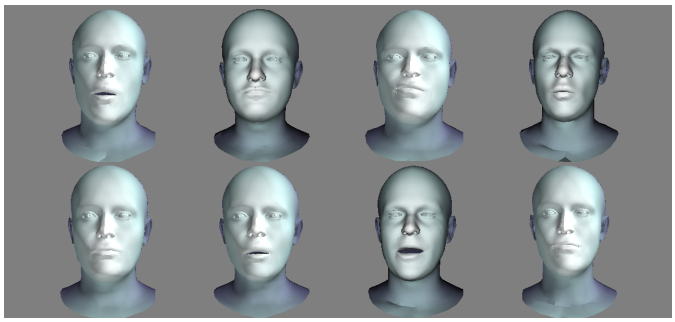

Figure 3: Generated meshes using $\widetilde{Di}$. The expressions above are sampled in the latent space from a normal distribution centered at 0 with $\sigma = 0.3$. All meshes are realistic and mix traits and expressions from various subjects.

We introduced a new decoder for a Variational Auto-Encoder which is able to learn fine meshes without intermediate graph coarsening, by combining spectral methods, residual nets and global pooling. New operators do not need preprocessing or high memory requirements and show a tangible improvement over the previous work, approaching recent benchmarks that employ graph down- and up-sampling. The resulting model is light, easy to train and deploy.

References

[1] M. Bronstein, J. Bruna,Y. LeCun,A. Szlam , P. Vandergheynst, “Geometric deep learning: going beyond euclidean data” IEEE SIG PROC MAG 2016.

[2] K. Crane, U. Pinkall, P. Schröder, “Spin transformations of discrete surfaces” ACM Transactions on Graphics (TOG) 2011.

[3] M. Defferrard, X. Bresson, P. Vandergheynst “Convolutional neural networks on graphs with fast localized spectral filtering” Advances in Neural Information Processing Systems 2016.

[4] K. He ,X. Zhang, S. Ren, J. Sun “Deep Residual Learning for Image Recognition” arXiv:1512.03385 2015

[5] D. P. Kingma, M. Welling, “An Introduction to Variational Autoencoders”arXiv:1906.02691 2019.

[6] I. Kostrikov, Z. Jiang, D. Panozzo, D. Zorin, J. Bruna, “Surface Networks” 2018 IEEE Conference on Computer Vision and Pattern Recognition 2018.

[7] A. Ranjan, T. Bolkart, S. Sanyal, M. J. Black, “Generating 3D faces using Convolutional Mesh Autoencoders” ECCV 2018.

[8] E. Tretschk, A. Tewari, M. Zollhfer, V. Golyanik, C. Theobalt “DEMEA: Deep Mesh Autoencoders for Non-Rigidly Deforming Objects” unpublished 2019.

[9] N. Vercheval, H. De Bie, A. Piurica, “Variational autoencoders without graph coarsening for fine mesh learning” submitted to ICIP 2020.