Deformation of images using Generative Adversarial Networks: a study to generate realistic novel images of the worm C. elegans

Roberto Ceraolo, Roberto Minini
Semester Project, Spring 2022, EPFL, Switzerland
Project hosted by Prof. Sahand Rahi, LPBS, Institute of Physics, EPFL

Introduction

The following is the report of the work done by Roberto Ceraolo and Roberto Minini in the context of their semester project in the Laboratory of the Physics of Biological Systems.

It is the second part of project that began during the Machine Learning course in the 2022-2023 Autumn semester. The results of the previous part of the project are available here (an EPFL account is needed).

In this second part, the aim of the work was still to obtain realistic deformations of images of C. Elegans. Nevertheless, this time, the images were not of neural activity but instead of the full shape of the body. The main reference for this work is the paper *Dimensionality and Dynamics in the Behavior of C. elegans* [2](the paper will be referenced as DBC hereinafter). The authors studied and created a quantitative model to describe the shape of C. Elegans. Indeed, considering the worm's curvature, they showed that the space of postures is low-dimensional and can be described by projecting on 4 principal components. The authors of the paper kindly provided us with a sample of the dataset they used for their publication. Our goal was to generate, through GANs [1], realistic deformations of their images. This way, in case we reached good results, we could use this architecture in applications in which data is not easy to retrieve.

We also re-built from scratch their quantitative methods to visualize the principal components of the postures, very closely to how as they are described in the paper. This allowed us to make comparisons between the distributions of generated and real images. Our code is available on request.

The logic we followed is the same as in the previous part of the project: starting by generating synthetic data resembling the real images, check the behavior of the GAN on them. Then, move to applying the GAN directly on real images. More or less halfway thorugh this project, we decided to change architecture of the GAN, from the one described in the previous report, to one based on StyleGAN2-ADA [4], which is one of the most recent and sophisticated architectures for image and video generations. The use of this different GAN brought to extremely better results with respect to the previous ones. While the previous one was struggling even with synthetic data, the new one is able to generate visually realistic deformations of the real C. Elegans images. They are hardly distinguishable from the real data by humans. When applying the quantitative procedure to compare the principal components of the matrix identifying the shapes, it seems that it still has some issues in replicating the distribution of shapes. Indeed, looking at the 2D graphs in 10 it seems that a part of the distribution is missing. Future work may address this issue and investigate what the missing parts have in common. When using the StyleGAN2-ADA, we checked how the quality and the distributions of images changed when training the GAN with sets of different sizes. It wasn't found a clear pattern, e.g., better results when increasing or decreasing the train set size. Indeed the sizes tested were 7200 (full dataset we were given), 4000, 2000 and 500 and the best results were given when training with 4k images.

I. Dataset Characteristics

A. The Data Set

The dataset was shared with us by the authors of DBC paper and it is a sample of the entire dataset they used for their paper. The sample we were given contained 7200 images of a Caenorhabditis elegans strain. For details of the conditions under which the images were captured the reader should refer to the original paper.

B. Realistic Deformations

As said, our final goal was to generate realistic deformations of the real images of the C. Elegans. In order to clarify, we need to define what we mean by realistic deformations. A realistic deformation is characterized as:

i focusing on the content, i.e. the body of the worm and not on the background
ii not distorting the content, i.e. not introducing artifacts
iii unique, i.e. not equal to other deformations
iv novel, i.e. different from all the samples in the training set

When such goal was reached for each of the images, we declare a deformation procedure complete if it manages also to reproduce the overall distribution of the shapes of the worms of the training set.

II. Terminology

In order to mantain clarity throughout the report, we use the following terms consistently:

i **DBC** = Paper from Stephens et al. describing the Dynamics in Behavior of C. Elegans (hence DBC)
ii **Real data** = Real images of C. Elegans, as provided from the authors of the DBC paper
iii **Synthetic images** = images generated by us using a script
iv **Generated images** = images generated by the GAN

III. Timeline

The following is a high-level timeline of the steps of the project.

i First of all, we started wondering: can the GAN learn to do a certain deformation on synthetic images? (swirl, rotation, etc.). Some deformation was working and some other was not.;
ii Then, we switched to asking ourselves: "Can the gan flex a straight line?". Here we tried sharply bent lines and sinusoidal ones. The GAN was struggling in doing it.;
iii From sinusoidal curves, we switched to Bezier curves with ellipsoids as "heads and tails" to emulate the shape of a worm. We wanted to understand whether the GAN could learn to deform in a realistic way the shape of a synthetic worm. Visually the result seemed decent, except for the small flaws on the border. So, we needed a quantitative measure. Could we say that the gan managed to reproduce the distribution of shapes of the worms in the training set? ;

iv Here we started using a quantitative procedure (described in IV-A) for the comparison of distributions of angles of the skeletons. It is the same that the authors of the DBC paper use. In brief, it produces graphs of the principal components of a matrix describing the shapes of the worms. We started by trying the procedure on our best deformations, which are synthetic images of worms generated with Bezier curves, generated with 2 flexing points. [1];

v Since part of the procedure was not working correctly, we generated "simpler" synthetic worms, with just 1 flexing point. With this modification, worms could only have arch-shapes and not "s"-shapes. We also implemented a different version of skeletonize that allowed for pruning when there were some flaws;

vi Result obtained: the distributions of the shapes of the generated and synthetic images were different. It seemed that the GAN was not able to replicate the distribution. We investigate more.

vii To further understand where the result obtained, which had a peculiar shape, we tried to generate images skewed on purpose: worms having curves only on the left or only on the right. Moreover, we also played with different degrees of freedom for the distance of the most extreme point of the curve from the vertical central axis of the image (both on the left and on the right). With these experiments we fully understood the graph: each of the two sides represents when the worm is curved on that side, and the points in the center represent situations in which the worm is very close to be vertical-shaped.

viii We also tried to change the eigenvectors onto which we projected the principal component in our procedure:

a) Projecting the principal components each of the two matrices (for synthetic and for deformed images) on their own eigenvectors

b) Projecting both on the eigenvectors of the synthetic images

c) Projecting both on the eigenvectors of the deformed images

ix Another hypothesis we made and tested was the following: maybe the difference in the graphs obtained for the principal components is explained by the fact that in the synthetic images the head and the tail move, whereas in the deformed they don't, since we are always deforming only one image.

x Finally, we tried to generate worms even more close to the ones from the real data, to check whether we could obtain the same graph as the authors of the DBC paper obtained on real data.

xi Even after all these experiments, the results were still not satisfactory. Hence, we decided to change the architecture of the GAN in favour of a StyleGAN2-ADA. The results of generated images when training the GAN on real data were visually very good.

xii We applied the quantitative procedure on the generated images obtained with the new architecture, and noticed that the GAN is able to partially reproduce the distirbution, but it still has some issues.

xiii To understand the quantity of data needed to generate good deformations, we ran the GAN with different training sizes.

[1]The flexing points are the points used to generate Bezier curves, following a certain formula. We use them in order to calibrate whether we want simpler or more "sinuous" lines.

IV. Method

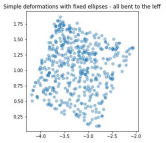
A. Quantitative analysis of distribution of shapes

To avoid relying only on visual judgment, we developed a way to compare the distributions of the shapes of a set of images of the worm. We used the same procedure that Stephens et al. used in their paper. For each image, we firstly binarize it and we crop it around the largest connected component (which always happens to be the worm). Then, we extract the skeleton of the worms by using a skeletonization function provided in the PlantCV library [3]. Sometimes the library produced skeletons which were not a single segment. For this reason, we had to also prune the skeletons, in order to remove the smallest segments of the skeleton. In order to do this, we used the pruning function in PlantCV, but since this was not working all of the times, we also implemented manual pruning. After having correctly skeletonized the pictures, we interpolate the spline based on the skeleton and we characterize the curvature of the worm by computing the angles of the curvature at each point of the spline. These angles, which we call thetas, are stored in a matrix. Based on these angles, we compute the PCA and we draw the charts and plots which we will show in section IV-C. To validate that we correctly reproduced the procedure in the DBC paper, we compared the graphs in the paper with those obtained through our procedure. You can find this comparison in the Appendix, VI-A4.

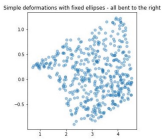
B. Before Style GAN

1) Investigations on bowtie (only synthetic images): Firstly, we applied the procedure to the synthetic images, in order to study the behaviour of the principal components and understand the high level meaning of each part of the pca graph obtained. By generating different datasets and taking note of the results obtained in each case, we were able to reach the above goal. When running the procedure on synhtetic images generated with Bezier curves with 1 flexing point, the pca yields what we called a *bowtie-shaped* graph 1a.

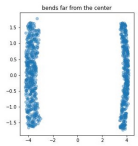
(a) Full Bowtie



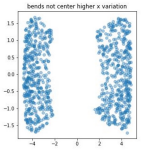
(b) Only to the left



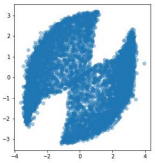
(c) Only to the right



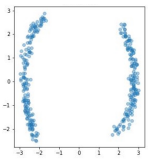
(d) Farther from medial axis



(e) Closer to medial axis



(f) More vertical-shaped curves



(g) Less vertical-shaped curves

Fig. 1: All of the bowties experiments

In order to fully characterize the bowtie graph, we ran several experiments:

i Bending the Bezier curves only to the left;
ii Bending the Bezier curves only to the right;
iii Making the curves further away from the vertical medial axis of the image;
iv Making the curves closer to the vertical medial axis of the image;
v Allowing more vertical-shaped curves (after having incremented the flexing points in order to better resemble the real data);
vi Allowing less vertical-shaped curves (after having incremented the flexing points in order to better resemble the real data);

Thanks to the above experiments, we understood that each side of it is due to the respective side of the curve, with positive values of the first principal component for the projected datapoints on the x axis for right curves and negative ones for left curves. Moreover, we proved that the points around 0 are reflected by the worms close to a vertical line. To do so, we generated only very curved worms, and we noticed missing points in the middle of the graph. The points in the center represent situations in which the worm is very close to be vertical-shaped.

2) Comparison of distributions (synthetic and generated): We then generated the deformations of the synthetic images with the architecture of the GAN and compared the pca graphs of the distributions of the images. In particular, we carried out two experiments:

i In the first one, we trained the GAN with synthetic images having 1 flexing point and the coordinates of head and tail allowed to move;
ii In the second one, we trained it with synthetic images generated with 2 flexing points and fixed positions for head and tail;

The following are the synthetic and generated images obtained in each of the two cases.

Fig. 2: Input dataset sample (1 flexing point)

Fig. 3: Generated images (from 1 flexing point training data)

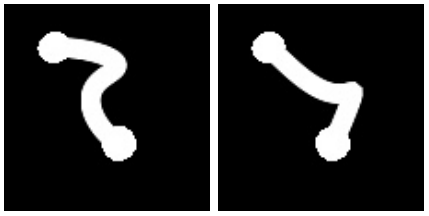


Fig. 4: Input dataset sample (2 flexing points)

Fig. 5: Generated images (from 2 flexing points training data)

The results are not satisfactory, nor visually, nor from a distribution point of view. Indeed, as can be seen, the shapes have some flaws on their contours. Also, when applying the quantitative procedure, we obtain very different graphs for the principal components of the matrix of angles.

3) Change of eigenvectors for projections: We then wondered whether the differences in the distributions of the shapes were due to the different eigenvectors onto which we were projecting in each of the two cases. Indeed, when plotting the distribution of real images, we were projecting the covariance matrix onto its own eigenvectors, and when considering the deformed images we were projecting onto the eigenvectors of the matrix of the deformed images themselves. To understand whether the differences between the distributions were due to this, we tried to do both projections onto the real images' ones and onto the generated images' ones. The following graphs are to show what happens when projecting on different eigenvectors.

The following is obtained when projecting both matrices onto their own eigenvectors, shown as a benchmark for comparison, in order to understand what happens when changing the eigenvectors.

i Each matrix projected onto its own eigenvector;
ii Both matrices projected onto the synthetic images' eigenvectors;
iii Both matrices projected onto the generated images' eigenvectors;

In this experiment the precise type of training data is not crucial, as we were investigating the effect of the change of the eigenvector projection. All the following graphs were obtained when considering the experiment i mentioned in the previous section. The GAN was hence trained on a dataset of synthetic images made with 1 flexing point and fixed positions for head and tail. The generated images were, as said before, more complex than the training dataset, as if they had 2 flexing points.

The results of the experiment done by changing the eigenvectors onto which the projection is done are shown in fig 6. It was clearly not the reason for which the two distributions differ.

C. After Style GAN

After we changed completely the architecture of the GAN, switching to the StyleGAN2 one, the results obtained improved dramat-

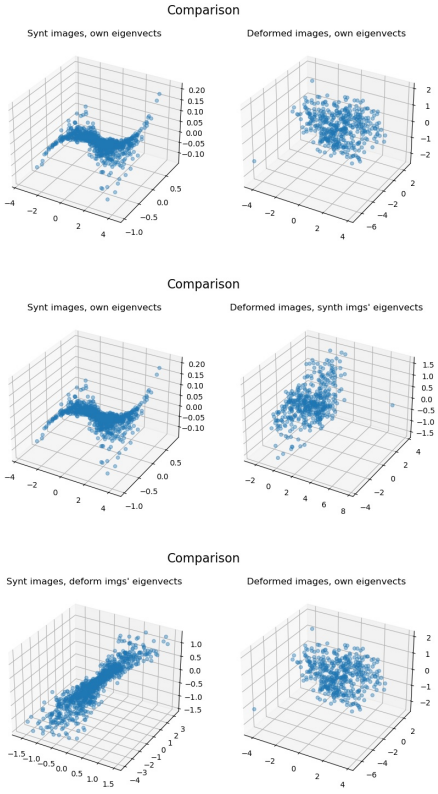


Fig. 6: All the eigenvectors experiments

ically. We tested the architecure on real data, using the real data and we obtained results visually very close to real data, not easily recognized by the human eye.

In order to verify whether the features of the dataset influenced the performance of the styleGAN, we run some different experiments. The two research questions we were investigating were:

i Can the GAN with this new architecture generate visually good images also with less than 7200 images? How does it behave when facing scarcity of data?;
ii Does the resolution of the training images have an effect on the ability of the GAN to generate deformations?;

Each time the GAN generated 7200 images, in order to compare their distribution of shapes with the full original dataset.

i Using the full real data dataset (7200 images) with each image of the training set having resolution 128x128;
ii Using a sample of 4000 images taken from the real data with each image of the training set having resolution 128x128;
iii Using a sample of 2000 images taken from the real data with each image of the training set having resolution 128x128;
iv Using a sample of 500 images taken from the real data with each image of the training set having resolution 128x128;
v Using a sample of 7200 images taken from the real data with each image of the training set having resolution 256x256;

Except for the last one, all the experiments yielded good visual results. In the experiment with images having resolution of 256x256, the GAN fails completely to generate deformations. In the other cases cases, the generated images are hardly distinguishable from the real data. In figure 7 you can see the generated images. We cannot show the real images here, but they can be seen in the DBC paper, and they are very similar to the generated ones shown here.

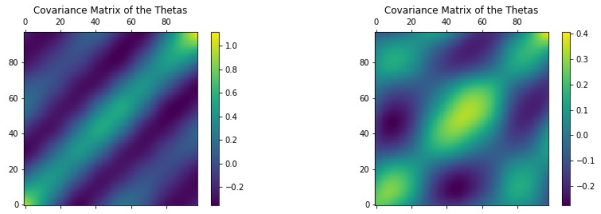
Fig. 7: Generated images using StyleGAN2-ADA and training set of 4k images

We applied our quantitative procedure in order to evaluate the quality of the deformations, and to check whether the GAN was able to reproduce the distribution of the shapes of the skeletons. We show in the appendix VI-A all the results obtained of each of the experiments. None of the graphs obtained through the pca has a precise correspondence to the one obtained on the training data. Nevertheless, the experiment that yielded results with the highest closeness to the training set one, is the one in which the training set comprised 4000 images. Below (images 7, 8, 9, 10) we show the graphs of comparison between the real images and on the 4k-trained GAN output.

The goal of the procedure outlined in section A was to understand whether the generative model can reproduce the distribution of the original pictures. To do so, we draw the charts which were proposed in the DBC paper. These charts are:

- Covariance Matrix: we compute the covariance matrix of the matrix containing the thetas.
- Plot of the variance explained by the first 4 eigenvectors
- PCA plot using two and three dimensions

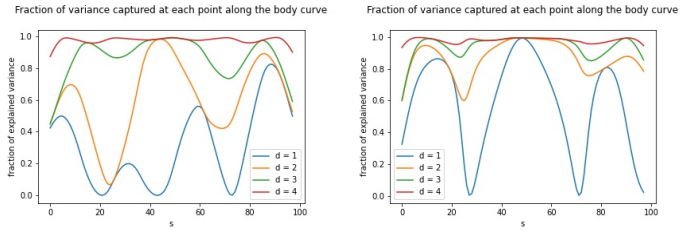
First we started by comparing the covariance matrices. As we can see in figure 8 the covariance matrices are somewhat different, since the one from the generated images seems to have a chessboard pattern.



(a) Original images (b) Generated images

Fig. 8: Covariance Matrix

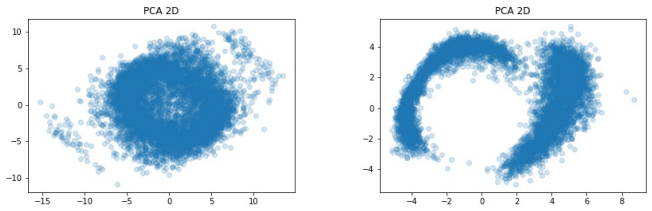
Then, we compared the explained variances plot. These graphs represent the variance explained at each point of the body (we discretized the skeleton by considering 100 points) by considering the first d eigenvectors, with d going from 1 to 4.



(a) Original images (b) Generated images

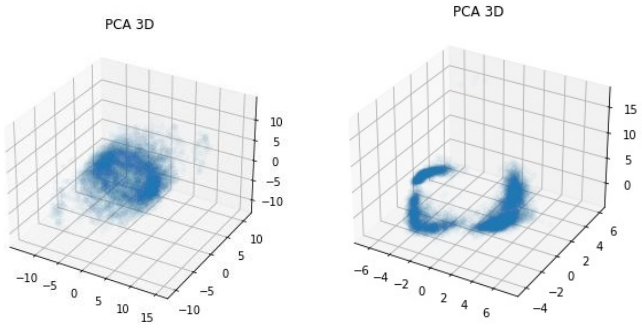
Fig. 9: Explained Variance

Finally using the first two and three dimensions of the projected data, we draw scatter-plots for the two-dimensional and three dimensional PCA.



(a) Original images (b) Generated images

Fig. 10: Comparison PCA 2D



(a) Original images (b) Generated images

Fig. 11: Comparison PCA 3D

As we can see that in figure 10 and 11 the original images case there is a clear circular shape which has its center approximately at 0. Instead, this shape is is not present in the generated images. However, we can still identify half of the circular shape. In other words, the GAN seems to be able to generate the part of the distribution.

V. Results and future work

To summarize, during our project we trained the GAN architectures on a great variety of images in order to understand where the model was working well and where the model was failing. After that, we focused on developing and implementing a procedure to compare the generated worms with the original ones. The main insights from the results of this project is that GANs, when built with the right architecture, are able to deform images of worms. We showed that the improved StyleGAN architecture manages to produce visually good deformations, that is deformations which are realistic and without imperfections. The StyleGAN is also partially able to emulate the original distribution of images, but not completely. Finally, we indicate some of the possible directions of future work that can be taken. Firstly, in addition to the comparison graphs we have provided, it would be interesting to implement numeric distance measures which are able to capture how close the distributions are. After that, some further research should be done on what kind of images from the original distribution the GAN is not able to deform. These are the ones which appear in the lower part of the circle in the DBC paper PCA graphs. We have tried to understand what these images have in common by visually inspecting them, but we could not find a definitive answer to this. After understanding what kind of deformations the GAN cannot generate, the next step is improving the GAN in order to produce the missing deformations. Once that is done, we will have a model which can produce satisfying images of the original DBC dataset.

References

[1] Ian J. Goodfellow et al. “Generative Adversarial Networks”. In: (2014). URL: https://arxiv.org/abs/1406.2661.

[2] Bethany Johnson-Kerner et al. Greg J. Stephens. “Dimensionality and Dynamics in the Behavior of C. elegans”. In: (2008). URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000028.

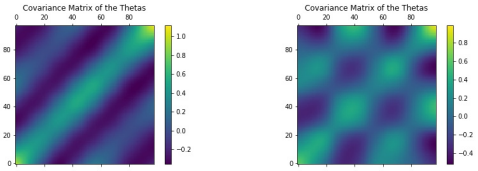
[3] Noah Fahlgren Malia Gehan. *PlantCV*. 2021. URL: https://plantcv.danforthcenter.org.

[4] Miika Aittala et al. Tero Karras. “Training Generative Adversarial Networks with Limited Data”. In: (2020). URL: https://arxiv.org/abs/2006.06676.

VI. Appendix

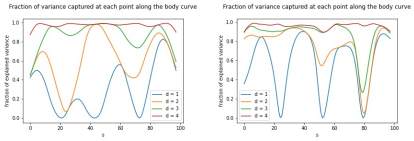
A. Experiments using Stylegan

1) GAN Trained on full dataset: The following is the comparison obtained when training using the full dataset.



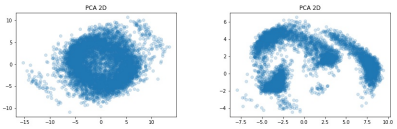
(a) Original images (b) Generated Images

Fig. 12: Covariance Matrix



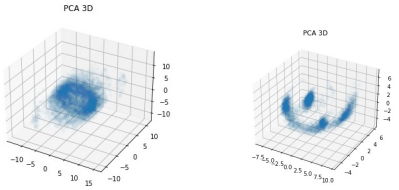
(a) Original images (b) Generated Images

Fig. 13: Explained Variance



(a) Original images (b) Generated Images

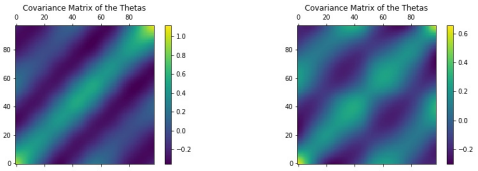
Fig. 14: Comparison PCA 2D



(a) Original images (b) Generated Images

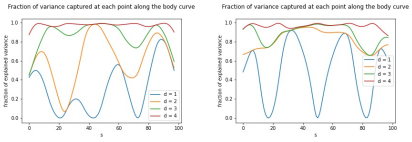
Fig. 15: Comparison PCA 3D

2) GAN Trained on 2000 images: The following is the comparison obtained when training using a subsample of 2000 images.



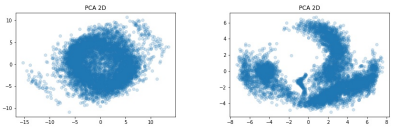
(a) Original images (b) Generated Images

Fig. 16: Covariance Matrix



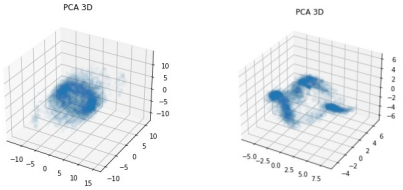
(a) Original images (b) Generated Images

Fig. 17: Explained Variance



(a) Original images (b) Generated Images

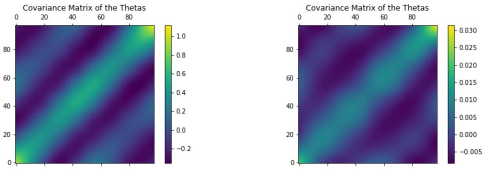
Fig. 18: Comparison PCA 2D



(a) Original images (b) Generated Images

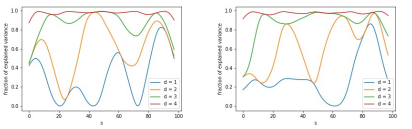
Fig. 19: Comparison PCA 3D

3) GAN Trained on 500 images: The following is the comparison obtained when training using a subsample of 500 images.



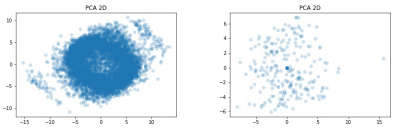
(a) Original images (b) Generated Images

Fig. 20: Covariance Matrix



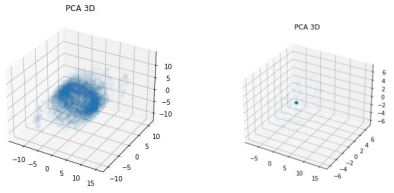
(a) Original images (b) Generated Images

Fig. 21: Explained Variance



(a) Original images (b) Generated Images

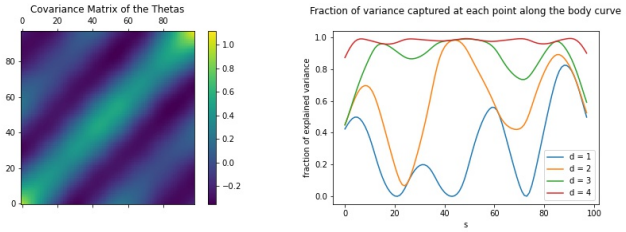
Fig. 22: Comparison PCA 2D



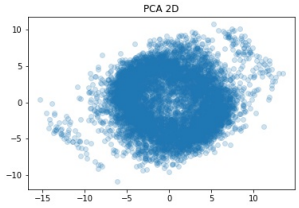
(a) Original images (b) Generated Images

Fig. 23: Comparison PCA 3D

4) Comparing the graphs obtained with our procedure to the ones obtained by the authors of the DBC paper: The following are the graphs obtained running our code on the real data, in order to doublecheck the correctness of the procedure. Indeed, we obtain results very similar to the ones in figure 2 of the DBC paper. Please note that the explained variances graphs are specular just because we had opposite labelling conventions for the points along the skeleton of the worm.



(a) Covariance matrix (b) Explained Variance



(c) 2D PCA

Fig. 24: Graphs produced with our procedure