Topology Layers for cGANs and DCGANs

Karhan Kaan Kayan
Department of Computer Science
University of Chicago
Chicago, United States
karhan@uchicago.edu

***Abstract*—Recently, it has been shown that it is possible to incorporate topological priors into Generative Adversarial Networks (GAN) using persistent homology and topological loss functions. Given accurate topological priors about the input distribution, generative models can converge to the population distribution much more rapidly by minimizing the topological noise. In this paper, we show that it is possible to apply this approach to Deep Convolutional GANs (DCGAN) and conditional GANs (cGAN). Empirical evidence in both architectures show that topological layers reduce the noise in the generated images to a significant degree. Furthermore, integrating these layers into conditional GANs makes it possible to promote different priors for different kinds of images in the dataset, which is not possible using GANs. Experimentally, we show that it is possible to get mild improvements over the original topology layer approach (using the same layer for every kind of image). We further test the topological layers in colored images with 3 color channels, and show that even though they do not improve the image quality, they might be of potential use in semantic segmentation tasks in the future.**

I. Introduction

In "A Topology Layer for Machine Learning" [1], it was shown that it is possible to incorporate topological priors to train better performing Generative Adversarial Networks. The authors first train their GAN model on a dataset, and then minimize the topological loss with respect to a loss function by (Adcock, Carlsson, and Carlsson, 2016) [2] using forward and backpropagation. The results indicate that it is possible to improve the clarity, quality, and the smoothness of the produced images using this idea. For the GAN experiments of the original paper, the topological priors mainly depend on 0-dimensional persistence features, which correspond to connected components. This is because even in a simple example such as handwritten digits, different kinds of images have different topological priors associated with them even in the same distribution, but 0-dimensional persistence penalty is typically the same for all kinds. For instance, images corresponding to digit 1 typically do not have holes in them, whereas the images corresponding to 6 do, but both would benefit from minimizing the noise around the edges due to disconnected components.

The aforementioned issue stems from the fundamental property of the Generative Adversarial Networks that they do not take into account auxiliary information in the input and only try to imitate the population distribution. To fix this issue a new architecture called conditional GAN [5] has been proposed to incorporate auxiliary information into the generator and the discriminator networks. This information is usually the labels of the images, which makes it possible to generate images with a desired label contrary to vanilla GAN. We show that it is possible to use topology layers with conditional GANs to promote different kinds of topological priors for different kinds of images. Our results show achieve mild improvements compared to the original approach.

We further test the extent of improvements of using topology layers on Deep Convolutional Generative Adversarial Networks (DCGAN). These are different from the GAN used in (Brüel-Gabrielsson et al.) [1] in that the generator and the discriminator use convolution layers instead of linear layers in their architecture. Our empirical results show that the topology layers do cause improvements in the quality of the images by reducing the topological noise, however the improvements are not as stark as in the case of vanilla GANs.

Finally, we test topology layers on colored images using three color channels. The results do not show any improvements over a standard GAN, which is possibly due to the capability of the GAN network used in this project. However, the topology layers produce an accentuation of image segments in certain cases, which is an evidence that they might be effective in semantic segmentation tasks.

II. Background

A. Topological Loss

We use the following topological loss function due to (Adcock, Carlsson, and Carlsson, 2016): [2]

Definition 1.

$$\mathcal{E}(p,q,i_0;\mathrm{PD}_k)=\sum_{i=i_0}^{|I_k|}|d_i-b_i|^p(\frac{d_i+b_i}{2})^q$$

This loss is defined for fixed p, q, i_0 and takes a k-dimensional persistence diagram PD_k as an input. For instance, since 0-dimensional homology counts the number of connected components, using $\mathcal{E}(p,q,2;\mathrm{PD}_0)$ penalizes all but the most persistent connected component in the data. This is because setting $i_0 = 2$ skips the most persistent feature in the sum. So, by using the loss function we can bias a machine learning algorithm to form only one connected component. Another example would be that using the loss function $\mathcal{E}(p,q,3,\mathrm{PD}_1)$ favors the formation of 2 holes in the data since 1-dimensional homology counts the number of holes. The parameter p can be increased to penalize the most persistent features more, and the parameter q can be increased to penalize features that appear later in the filtration.

The idea behind being able to integrate this loss function into a neural network is that it is differentiable, and therefore we can perform backpropagation on it. It's derivative with respect to a simplex σ is the following:

$$\frac{\partial \mathcal{E}}{\partial \sigma} = \sum_{i \in I_k} \frac{\partial \mathcal{E}}{\partial b_i} \mathbb{I}_{\pi_f(k)(b_i)=\sigma} + \sum_{i \in I_k} \frac{\partial \mathcal{E}}{\partial d_i} \mathbb{I}_{\pi_f(k)(d_i)=\sigma}$$

where $\pi_f(k)$ maps birth and death of a homology class to the simplex that created and destroyed it.

B. Generative Adversarial Networks

Generative Adversarial Nets (GAN) [3] are frameworks to estimate generative models in Machine Learning. They are implemented by training two separate neural networks, a generator network, and a discriminator network. The objective of the generator is to try to estimate an unknown probability distribution given samples from that distribution. For instance, if the distribution is the images of human faces, then the generator tries to generate realistic human face images. The objective of the discriminator is to distinguish real samples from the samples generated by the generator. In the previous example, a discriminator would try to distinguish between real human faces and fake ones generated the neural net.

The topological layer idea is very suitable for this kind of problem. In "A Topology Layer for Machine Learning" [1], the authors demonstrate that it is possible to denoise handwritten digits generated by a GAN by adding a topology layer to the GAN. The persistence diagrams for the topological loss are calculated using the superlevel set filtration on the pixel values. The idea relies on the observation that all based on this filtration, there should be 1 connected component for each digit. So, it is possible to promote 1 connected component using the loss function $\mathcal{E}$ with 0-dimensional persistence PD_0 to penalize the formation of more than 1 connected components. The authors demonstrate that the topological GAN trained for much less number of iterations performs reasonable well compared to a vanilla GAN trained for longer.

1) Deep Convolutional GANs: GANs are usually used to generate and learn images. And the most important aspect of generating images from a distribution is the architecture of the neural networks that correspond to the generator and the discriminator. For this reason, a much more effective version of GANs are usually used on image generation tasks called DCGANs. DCGAN stands for deep convolutional generative adversarial network, and uses Convolutional Neural Networks to model the generator and the discriminator. It has been introduced in (Radford, Alec, Luke Metz, and Soumith Chintala, 2016) [4] and has been shown to be much a much more effective way to generate images compared to vanilla GANs.

2) Conditional GANs: Vanilla GANs are state-of-the-art generative models for images, however they lack the ability to incorporate auxiliary information into the model. For instance, it is not possible to generate a human face with a particular trait with a vanilla GAN trained on pictures of human faces. To overcome this obstacle, (Mirza, Mehdi, and Simon Osindero, 2014) [5] introduced conditional GANs, cGANs for short. cGANs embed auxiliary information of the training data to the generator and the discriminator using an embedding layer at the beginning of the neural networks. This advancement makes it possible for GANs to generate images with given labels, given that the input set is labeled. cGANs have been recently shown to be useful for Image-to-image translation [6], Text-to-image synthesis, Video generation [7], and image segmentation [6].

Fig. 1: Before Topological Layer

Fig. 2: After Topological Layer

Fig. 3: Vanilla GAN results with and without the use of topological loss

III. Topological vanilla GAN

Following the code provided in (Brüel-Gabrielsson et al.) [1], I first reproduced the results of this paper to see that it is, indeed, possible to reduce topological noise using a topological layer in vanilla GANs. This GAN uses 4 linear hidden layers of size 128, 256, 512, and 1024 respectively. The topological loss is calculated using the sum of the persistence feature lengths of dimension 0. The persistence diagram is computed with the Dionysus library for TDA using superlevel set filtration of the Freudenthal triangulation of the image. Gradient descent was performed using the PyTorch library. [8]

We can see in figure 3 that the topological noise function did cause the generated images to be clearer. This is what we have expected since penalizing connected components should

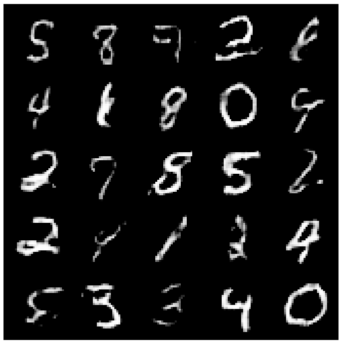

Fig. 4: Before Topological Layer

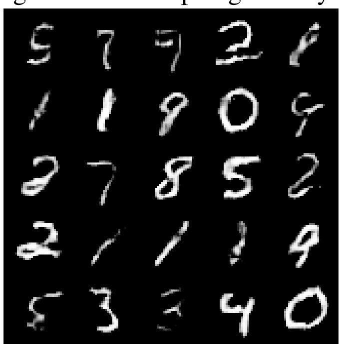

Fig. 5: After Topological Layer

Fig. 6: DCGAN results with and without the use of topological loss

incentivize one connected component, which produces clearer images.

IV. Topological DCGAN

To see the effect of a topology layer in reducing the topological noise of fake images generated by a DCGAN, I programmed and trained a DCGAN on MNIST handwritten digits dataset and the Fashion MNIST dataset, which is a more complicated dateset. The architecture for the generator consisted of a fully connected layer and 3 convolution layers with kernel size of 3, stride 1, and padding 1. Batch normalization was used in between every layer, and as the activation function, Leaky ReLU was used between all hidden layers. Finally tanh was used to convert the neural network output to pixel value. I tried 3 different topological loss functions to minimize the topological noise produced by the DCGAN. These were all calculated based on the persistence diagram induced by the superlevel set filtration of the Freudenthal triangulation of the images. The first loss function was the sum of the length of persistence pairs in the computed persistence diagram. The other two loss functions were $\mathcal{E}(1, 0, 1, \mathrm{PD}_0)$, and $\mathcal{E}(2, 0, 1, \mathrm{PD}_0)$, chosen with the goal of minimizing the number of connected components formed. These have different values for p, which means that they penalize more prominent persistence features differently. The DCGAN was trained for 20 epochs on the MNIST handwritten digits dataset and for 30 epochs on the Fashion-MNIST dataset. The trained models were then fed through the Adam optimization algorithm to

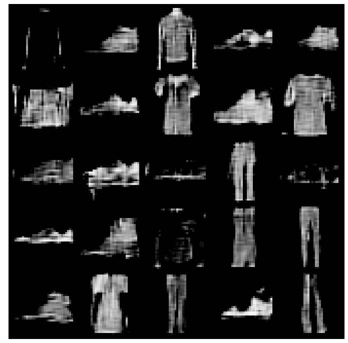

Fig. 7: Before Topological Layer

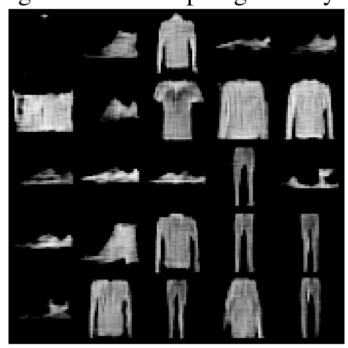

Fig. 8: With loss function as the sum of the length of the 0 dimensional persistence pairs

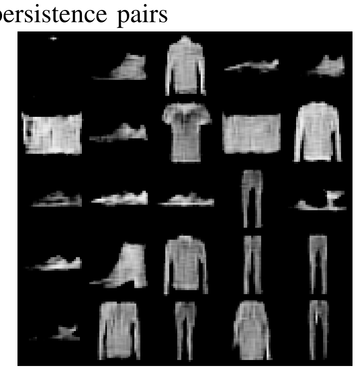

Fig. 9: With loss function as $\mathcal{E}(1, 0, 1, \mathrm{PD}_0)$

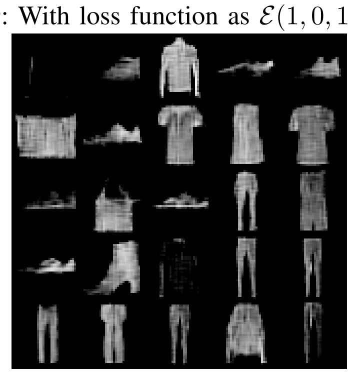

Fig. 10: With loss function as $\mathcal{E}(2, 0, 1, \mathrm{PD}_0)$

Fig. 11: DCGAN results on the Fashion MNIST dataset with and without the use of topological loss

minimize the topological loss for 20 epochs using Stochastic Gradient Descent.

For the MNIST handwritten digits data, we see in figure 6 that minimizing topological loss clearly produced better results. This worked by removing the connected components that correspond to noise as can be seen from the 8 figure in the middle, 2 figure on the right center and the 5 figure on the bottom left for instance. It also connected the previously disconnected parts and produced smoother images as can be seen for instance from the bottom 0 and 3 and top left 5.

For the Fashion-MNIST data, we see in figure 11 that the results with this dataset were even better compared to handwritten digits. We see that the fundamental problem with the DCGAN was that it did not fill the images into one piece. This obstacle was overcome by the topological loss function by penalizing the formation of multiple connected components. We also observe that the topological DCGAN produced smoother images around the edges by removing the noise around the edges. When it comes to the choice of the loss function, it seems like the sum of persistence pairs of dimension 0 and $\mathcal{E}(1, 0, 1, \mathrm{PD}_0)$ produced the most clear images.

V. Topological Conditional GAN

In this section, I wrote and trained a cGAN architecture on the MNIST handwritten digits and the Fashion MNIST datasets. The neural network for the generator consisted of the embedding layer for the labels, 4 hidden layers of sizes 128, 256, 512, 1024, respectively and tanh activation function to convert the output to pixel value. Between every hidden layer, batch normalization was performed and neurons were activated through Leaky ReLU layers.

It is not possible to integrate different topological priors into unconditional GANs because we are not able to control the labels or the auxiliary information of the generated images. So, we are stuck with using a single topological loss, which may or may not be appropriate for different kinds of images. In theory, this should be possible with conditional GANs since they are able to embed auxiliary information and labels into the generator model. Therefore, the main hypothesis that I wanted to test with this experiment was that incorporating different topological loss functions tailored to different kinds of images produces better results than minimizing a single loss function as was done in the original topology layer paper [1] and previous implementations in this project.

The experimental setups were as follows: I first trained the cGAN model on the handwritten digits conditioned on their labels for 20 epochs. Then, I tested two different topology layers on the generator model. The first one was a standard 0-dimensional persistence loss function $\mathcal{E}(2, 0, 1; \mathrm{PD}_0)$. The idea with this loss function was to reduce topological noise by penalizing the formation of different connected components. The other loss function was tailored specifically for each digit depending on its topological properties. The loss function corresponding to each digit was as follows:

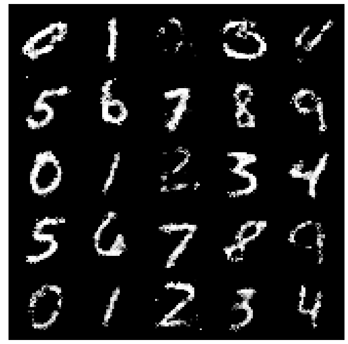

Fig. 12: Before Topological Layer

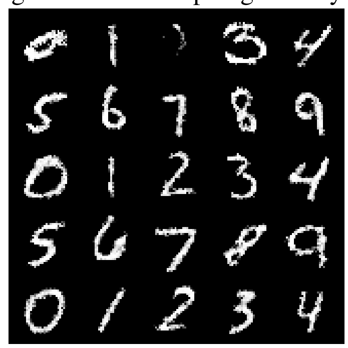

Fig. 13: With $\mathcal{E}(2, 0, 1; \mathrm{PD}_0)$ as the loss function

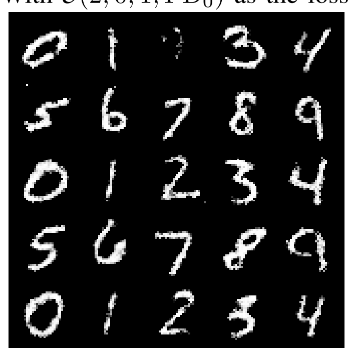

Fig. 14: Specialized loss functions for each digit

Fig. 15: cGAN results on MNIST digits with and without the use of topological loss

- Numbers 0, 6, and 9 have only one connected component and 1 hole. The corresponding loss was $\mathcal{E}(1, 0, 2; \mathrm{PD}_0) + \mathcal{E}(1, 0, 2; \mathrm{PD}_1)$.
- Numbers 1,2,3,4,5,7 have only one connected component and no holes. The corresponding loss was $\mathcal{E}(1, 0, 2; \mathrm{PD}_0)$.
- Number 8 has one connected component and two holes. The corresponding loss was $\mathcal{E}(1, 0, 2; \mathrm{PD}_0) + \mathcal{E}(1, 0, 3; \mathrm{PD}_1)$.

I trained both topological layers separately for 20 epochs. The results are shown in figure 15. First, note that I was able to produce images of different labels in order since these are produced by a conditional GAN. This was not possible using a normal GAN. Looking at the images, we can see that compared to the baseline first figure, both topological layers produced much clearer and smoother images. The effect of the specialized loss functions was not as strong as I had hoped. We can see clear good results such as the top left 0 digit, which has a much more defined hole in the third figure compared to the first two. The digit 8 is also a good example since its holes are more clear in the case of specialized loss function. However, these good results did not hold for all digits. In general, it seems to me that some digits in the second figure with the fixed loss function had less noise around the edges. The digits 4 and 2 were examples of this. This could possibly have to do with the fact that when we try to minimize multiple loss functions, we take away from our ability to minimize a single one of them optimally. So, the topological GAN with specialized losses might not have minimized the 0-dimensional persistence loss as much as the standard topology layer, and thus producing less smoother digits in some cases. On the other hand, the results with the digits 8 and 0 indicate that it might be optimal to mix these two approaches to get a better topology layer.

The experiment with the Fashion-MNIST dataset was different in the sense that I did not test for the same hypothesis as all of the topological priors on different labels were the same. Every kind of image in this dataset contains one connected component and zero holes. I first trained the cGAN for 50 epochs conditioned on the labels. Using the generator obtained, I trained 3 different topology layers corresponding to 3 different topological loss functions. The first one was the sum of the lengths of persistence pairs of dimension 0. The second one was $\mathcal{E}(1,0,1;\mathrm{PD}_0)$. The third one was $\mathcal{E}(2,0,1;\mathrm{PD}_0)$. So, the last two loss functions differed in the p parameter.

The results of this experiment can be seen in figure 20. All three topology layers produced much better results than the original cGAN. All of them improved the baseline much more than they did in the handwritten digits experiment. The reason for this might be that in the case of Fashion-MNIST, the noisy parts of the image are typically surrounded with points that belong to the same connected component, which is not the case with handwritten digits. For this reason, it should be easier to remove the noise in Fashion-MNIST using 0 dimensional persistence loss functions. When it comes to the loss functions, we would expect for a $\mathcal{E}(2,0,1;\mathrm{PD}_0)$ to be incentivized to merge two prominent connected components into one more than $\mathcal{E}(1,0,1;\mathrm{PD}_0)$ as higher p penalizes prominent persistence features more. This expectation is confirmed by observing the boot figure on the right. However in general, $\mathcal{E}(1,0,1;\mathrm{PD}_0)$ and sum of barcodes did a better job at producing more full, clear, and smoother images.

Finally, I attempted to test the topological layer method on colored images. I trained the cGAN on CIFAR-10, a dataset of colored images with 10 labels. The training phase took 80 epochs. The resulting generator was then used to feed 3 different topology layers, each corresponding to different topological loss functions. As in the previous experiment, the first one was the sum of the lengths of persistence pairs of dimension 0. The second one was $\mathcal{E}(1,0,1;\mathrm{PD}_0)$. The third one was $\mathcal{E}(2,0,1;\mathrm{PD}_0)$. This time, the process for minimizing

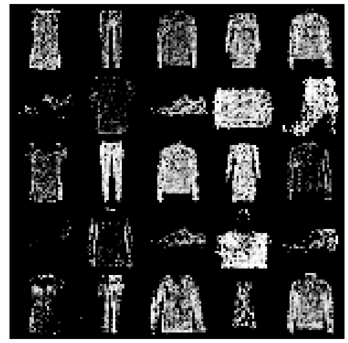

Fig. 16: Before Topological Layer

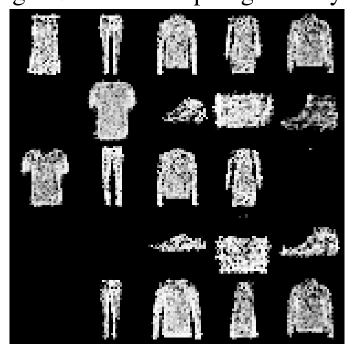

Fig. 17: With loss function sum of lengths of persistence pairs of dimension 0

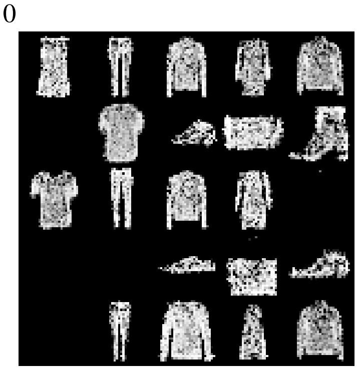

Fig. 18: With $\mathcal{E}(1,0,1;\mathrm{PD}_0)$ as the loss function

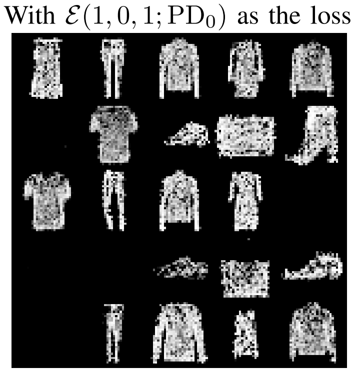

Fig. 19: With $\mathcal{E}(2,0,1;\mathrm{PD}_0)$ as the loss function

Fig. 20: cGAN results on Fashion-MNIST with and without the use of topological loss

loss was different since the images are colored. For each image, I minimized the losses on each of the 3 channels separately.

The results are shown in figure 25. Because of the lack of computational resources the cGAN has not been trained for long enough. It can be seen from the first figure that the generator did not successfully model the input distribution. However, I still wanted to see the effect of topological noise reduction on all color channels to figure out if that would make the images less blurry, clearer, or smoother. In all three topological layers, the images got either much darker, or much brighter. In all cases, topological layers accentuated certain segments of the image by removing the topological noise around them. The two bright images in the bottom line are a good example of this. If similar colors are thought of as belonging to the same image segment, by removing the noise around the segment, the topological layer can be thought of as improving the segmentation quality of the image. Although, the topology layers have not improved the quality of the produced images, it was effective in improving the segmentation of the image. This, might show that topology layers can be effective in semantic segmentation tasks, even though this experiment was not directly a semantic segmentation task.

VI. Conclusion

In this project I first replicated the results for topological GANs in (Brüel-Gabrielsson et al.) [1] to confirm that incorporating topological priors into GANs improve their quality. I further hypothesized that by using conditional GANs, we can incorporate different topological priors for different kinds of images and improve the produced image quality over standard GANs and the standard topological GANs. This hypothesis was tested on the MNIST handwritten digits datasets and was shown to be true to a small extent. The reason why the improvements were not as stark as I had hoped might be that in the case of 2-D images tested in this paper, most of the errors and topological noise stem from disconnected components, which are readily remedied by the standard topology layer approach. The conditional topological GAN approach used in this paper might give better results in 3-D datasets where 1 and 2 dimensional persistence features are more important.

The topological layer idea was further tested on various datasets, using different topological loss functions, and different GAN architectures, such as DCGAN, and cGAN. The results showed that except in the case of colored images, topology layers improved the quality of generators on every dataset. The improvements were much better on the datasets where there is a large noticeable segment and on architectures that use linear layers instead of convolution layers. The reason for this result is possibly that DCGANs are already designed to produce smooth images using convolution layers, which is similar to the purpose of adding topology layers. So, if topology layers are incorporated into GANs, it might be a better idea to incorporate them into Vanilla GANs instead of DCGANs to get better improvements.

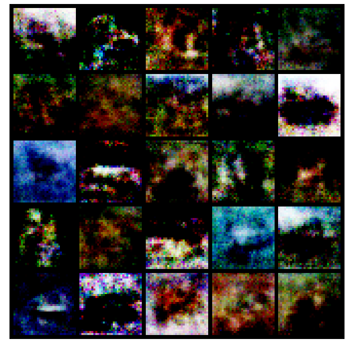

Fig. 21: Before Topological Layer

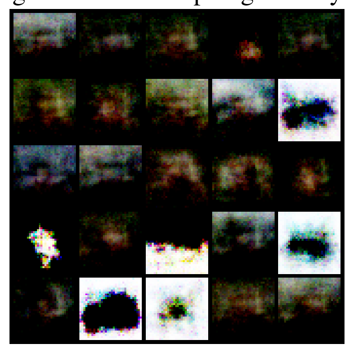

Fig. 22: With loss function sum of lengths of persistence pairs of dimension 0

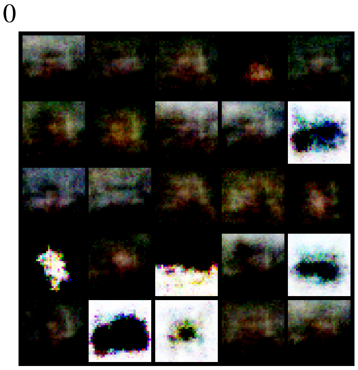

Fig. 23: With $\mathcal{E}(1, 0, 1; \mathrm{PD}_0)$ as the loss function

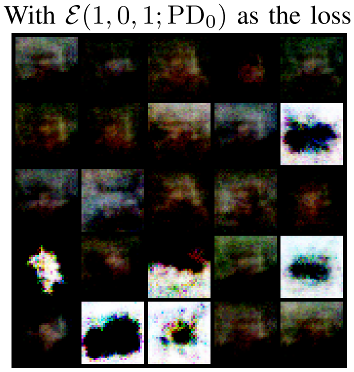

Fig. 24: With $\mathcal{E}(2, 0, 1; \mathrm{PD}_0)$ as the loss function

Fig. 25: cGAN results on CIFAR-10 with and without the use of topological loss

Finally, it was tested whether or not topology layers would improve the quality of colored images produced by cGANs. The experiments showed that they did not improve the quality and produced a certain darkening and brightening effect on the images. This might be due to the fact that the baseline generator was not that successful in modeling the original distribution, which left too much topological noise for the topology layer to handle. This could possibly indicate that the topology layers might be counterproductive when used above a certain topological noise level. Despite, these negative results, however, it was noted that topology cGANs accentuated prominent image segments in certain cases. This might show that the topology layer idea can be used in semantic segmentation tasks to reduce the noise around the segments.

References

[1] Brüel-Gabrielsson, Rickard, Bradley J. Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J. Guibas, and Gunnar Carlsson. *"A Topology Layer for Machine Learning."* . ArXiv:1905.12200 [Cs, Math, Stat], April 24, 2020. http://arxiv.org/abs/1905.12200.

[2] Adcock, Aaron, Erik Carlsson, and Gunnar Carlsson. *"The Ring of Algebraic Functions on Persistence Bar Codes."* Homology, Homotopy and Applications 18, no. 1 (June 2016): 381–402. https://doi.org/10.4310/HHA.2016.v18.n1.a21.

[3] Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio.

[4] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks." ArXiv:1511.06434 [Cs], January 7, 2016. http://arxiv.org/abs/1511.06434.

[5] Mirza, Mehdi, and Simon Osindero. "Conditional Generative Adversarial Nets." ArXiv:1411.1784 [Cs, Stat], November 6, 2014. http://arxiv.org/abs/1411.1784.

[6] Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. "Image-to-Image Translation with Conditional Adversarial Networks." ArXiv:1611.07004 [Cs], November 26, 2018. http://arxiv.org/abs/1611.07004.

[7] Villegas, Ruben, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. "Decomposing Motion and Content for Natural Video Sequence Prediction." ArXiv:1706.08033 [Cs], January 7, 2018. http://arxiv.org/abs/1706.08033.

[8] Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." ArXiv:1912.01703 [Cs, Stat], December 3, 2019. http://arxiv.org/abs/1912.01703.