Persistent Homology Based Generative Adversarial Network

Jinri Bao[1,2], Zicong Wang[1,2], Junli Wang[1,2] and Chungang Yan[1,2]
[1] *Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, Shanghai, China*
[2] *National (Province-Ministry Joint) Collaborative Innovation Center for Financial Network Security, Tongji University, Shanghai, China*

Keywords: Generative Adversarial Network, Persistent Homology, Topological Feature.

Abstract: In recent years, image generation has become one of the most popular research areas in the field of computer vision. Significant progress has been made in image generation based on generative adversarial network (GAN). However, the existing generative models fail to capture enough global structural information, which makes it difficult to coordinate the global structural features and local detail features during image generation. This paper proposes the Persistent Homology based Generative Adversarial Network (PHGAN). A topological feature transformation algorithm is designed based on the persistent homology method and then the topological features are integrated into the discriminator of GAN through the fully connected layer module and the self-attention module, so that the PHGAN has an excellent ability to capture global structural information and improves the generation performance of the model. We conduct an experimental evaluation of the PHGAN on the CIFAR10 dataset and the STL10 dataset, and compare it with several classic generative adversarial network models. The better results achieved by our proposed PHGAN show that the model has better image generation ability.

1 INTRODUCTION

Image generation has always been a key problem in the field of computer research, and how to make computers automatically generate realistic images has always puzzled computer scholars. In 2014, the emergence of Generative Adversarial Network (GAN)(Goodfellow et al., 2014) made significant progress in computer-generated images. GAN consists of a generator and a discriminator. The generator is trained to generate images that are as similar as possible to real images, and the discriminator is trained to determine whether the potential distribution of the generated images is consistent with the potential distribution of the real images. Through the confrontation between the generator and the discriminator, the generator can generate more and more realistic images.

After GAN was proposed, Radford et al.(Radford et al., 2015) conducted further research on the underlying architecture of GAN and used the convolutional neural network as the underlying architecture of the generator and discriminator of GAN. The generation performance has been greatly improved, leading to commonly use of convolutional neural network in subsequent GAN-based models(Zhu et al., 2017; Wang et al., 2022) as backbones.

Although GANs based on the convolutional neural network structure have achieved success in image generation, there are still some problems to be solved: the model performs well in generating local details but poorly in overall structure generation. Study(Zhang et al., 2019) found that this is because GANs based on the convolutional neural network rely on the convolution operation for feature extraction, while the convolution operation has a limited size of the convolution kernel. Its receptive field is limited, and some long-distance dependencies cannot be captured, so that the model does not perform well in the overall structure.

In recent years, persistent homology (PH)(Zomorodian and Carlsson, 2004) has attracted attention in terms of data feature extraction. Compared with existing data feature extraction methods, persistent homology method can connect algebra and topology, and provides measurable global information. The quantitative numerical value of the topological feature has opened up new research directions in the field of computer science. For example, Kindelan et al.(Kindelan et al., 2021) and Khramtsova et al.(Khramtsova et al., 2022) researched the classification problem based on persistent homology. Byrne et al.(Byrne et al., 2022) and Li et al.(Li et al., 2022) researched the image

Bao, J., Wang, Z., Wang, J. and Yan, C.
Persistent Homology Based Generative Adversarial Network.
DOI: 10.5220/0011648200003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 4: VISAPP*, pages 196-203
ISBN: 978-989-758-634-7; ISSN: 2184-4321

segmentation based on persistent homology. Carriere et al.(Carrière et al., 2020) and Kim et al.(Kim et al., 2020) proposed the topological feature layer that can be embedded in neural networks based on persistent homology. Moor et al.(Moor et al., 2020) studied the optimization of machine learning models using the topological features of the data as a new loss term.

This paper proposes the Persistent Homology based Generative Adversarial Network (PHGAN). Based on the original convolutional neural network architecture of GAN, the topological features obtained by persistent homology are integrated into GAN. The topological features of the data make up for the lack of the original model's ability to capture long-distance dependencies so that PHGAN can coordinate the global structural features and local detail features when generating images.

2 RELATED WORK

2.1 Persistent Homology

The topological features obtained by persistent homology are generally represented by the persistent diagram or the persistent barcode. However, these data formats are not suitable for subsequent machine learning tasks, so some researches are carried out on topological feature transformation. Adams et al.(Adams et al., 2017) and Cang et al.(Cang et al., 2018) researched how to transform topological features into two-dimensional matrices or three-dimensional tensors. After transformation, such data formats can be treated as images for machine learning tasks. Mileyko et al.(Mileyko et al., 2011) proposed to use the Wasserstein distance to measure the proximity of the topological features of the two data. Merelli et al.(Merelli et al., 2015) proposed to use entropy to measure the distribution of topological features and assign a certain entropy value to the distribution of topological features of data. Hofer et al.(Hofer et al., 2017) proposed to use of a neural network for topological feature transformation so that the neural network can learn to obtain topological feature transformation parameters that are most suitable for machine learning tasks.

In this paper, we proposed to transform topological features obtained by persistent homology into one-dimensional vector, then the vector can be input into neural network for processing.

2.2 Persistent Homology Based Generative Model

Recently, some researchers have explored the application of persistent homology in the field of image generation. For example, Khrulkov et al.(Khrulkov and Oseledets, 2018) used the approximate value of the topological features of the generated images and the real images as an indicator to measure the generation performance of GANs. Coincidentally, Horak et al.(Horak et al., 2021) also proposed a different GANs generation performance evaluation index based on the persistent homology method. However, what they proposed only used the topological features of the generated images and real images to measure the generation performance of GANs, and did not use the topological features of the real images to guide the generator to generate images.

Brüel-Gabrielsson et al.(Gabrielsson et al., 2020) proposed to use the topological features obtained by persistent homology to guide GAN to generate images, but the author only did explicit topological feature optimization for the noise input to the generator, and the generator did not learn the topological feature distribution of real images.

In addition, there are also some studies on the application of persistent homology in other generative models. For example, Schiff et al.(Schiff et al., 2022) proposed a variational autoencoder model based on the persistent homology, using the topological features as a new reconstruction loss term to optimize the generation performance of the variational autoencoder model.

Our proposed PHGAN uses the topological features of real images to guide the generator of GAN to generate images, so that the generator can learn the topological feature distribution of real images.

3 METHOD

The overall model architecture of our proposed PHGAN is shown in Figure 1. We sample random noise from a Gaussian distribution, and then feed this noise into the generator to generate an image. The generated image and the real image are input into the discriminator to discriminate the real and fake. In the discriminator, the input image is not only processed by the convolution module to obtain the features of the convolutional neural network, but also processed by the persistent homology module and topological feature transformation module to obtain the topological features. These two features are connected in series to discriminate the real and fake images.

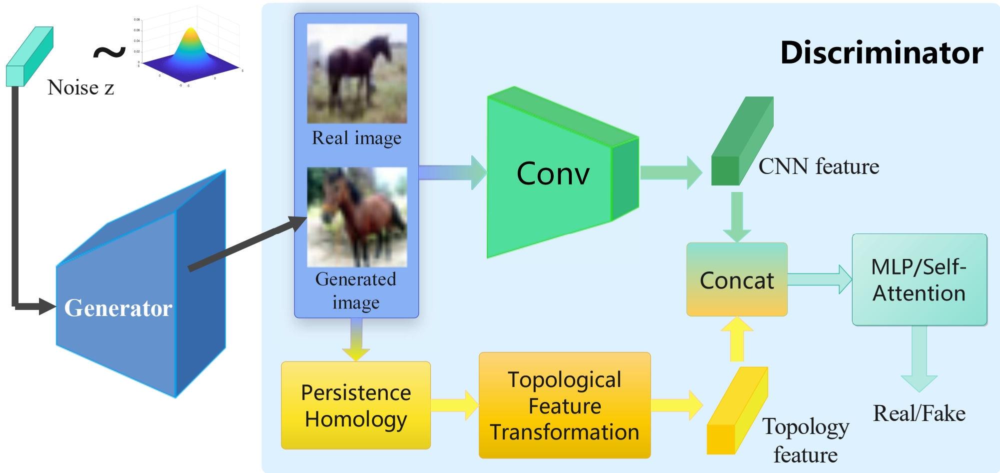

Figure 1: PHGAN Architecture.

In this way, we can incorporate the topological features which reflect the global structure of the image into GAN. On the one hand, the topological features enable the model to discriminate the real and fake images on the global structure, and on the other hand, in the adversarial training process of the generator and the discriminator, the ability of the generator to generate images can be improved. The specific implementation details are described below.

3.1 Persistent Homology: From Image to Topological Features

Persistent homology is a method for extracting the topological features of data. The basic process is to construct complex and complex filtering based on the original data, and then extract the topological features of the data. In this paper, the datasets consist of only two-dimensional images, and for image data, cubical complex(Ziou and Allili, 2002) is the most suitable choice for complex construction. We represent the two-dimensional image data with a two-dimensional array X of $N_i * N_j$, the value of each point X_{ij} in the array is the value of the image pixel, and then we construct a subset of the array X, that is, the set of pixels in the array X that are below the threshold t, as shown in Eq. (1). We use the S to denote a subset of the array X.

$$S(t) = U_{i,j} X_{ij} : X_{ij} <= t \tag{1}$$

where U denotes the set of pixel points.

When the threshold t changes from small to large, we can get a series of sets:

$$\varnothing \subseteq S(0) \subseteq S(t_1) \subseteq S(t_2) \subseteq \ldots \subseteq S(1) \subseteq X \tag{2}$$

Each such set of pixel points can be constructed to form a cubical complex, and the cubical complex formed by this series of sets is called a complex filtering.

When the value of t is relatively small, according to Eq. (1), the set S consists of only a few pixels. As the threshold t continues to increase, new pixels are added to the set S to form a new cubical complex, and the topological features appear and disappear during the transformation of the old and new cubical complex. The persistent homology method is to calculate the number of topological features of the cubical complexes formed under different thresholds. We use β_k to represent the number of topological features of k-dimension: β_0, the number of topological features of 0-dimension (connected components); β_1, the number of 1-dimensional topological features (rings/holes). Because we are studying two-dimensional image data, we only involve 0-dimensional and 1-dimensional topological features here.

The final result of the persistent homology method is the appearance and disappearance of each topological feature (appears at the threshold t_{birth} and disappears at the threshold t_{death}). We generally use a persistent diagram or a persistent barcode to represent the result of persistent homology analysis, as shown in Figure 2.

For topological features, the longer the persistent time (the disappearance time minus the appearance time we call the persistent time), the more important and meaningful the feature is. If topological features are of short persistent time, we usually treat them as noise.

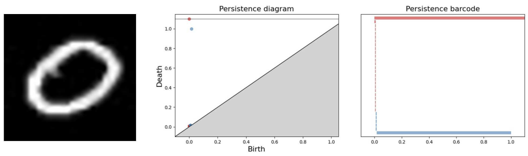

Figure 2: Persistent diagram and persistent barcode; the far left is 0 in the MNIST dataset. After persistent homology analysis, 0-dimensional topological features (connected components, red) and 1-dimensional topological features (rings/holes, blue) are obtained.

3.2 Topological Feature Transformation

After the image is processed by the persistent homology module, the obtained topological features are expressed as a persistent diagram or a persistent barcode. However, these data formats are not suitable for input into the subsequent discriminator. Therefore, we need to transform the topological features.

We use the persistent time of each topological feature as a measure of this topological feature:

$$\tau_k^i = d_k^i – b_k^i \tag{3}$$

where b_k^i, d_k^i, τ_k^i represent the appearance time, disappearance time and the persistent time of the i-th k-dimensional topological feature.

For our image data, the topological features obtained by persistent homology analysis have two dimensions, one of which is the 0-dimensional connected components, and the other is the 1-dimensional holes. We combine the persistent time of 0-dimensional and 1-dimensional topological features contained in an image to form a vector. In this way, the topological features of the image are transformed into a vector data format.

The specific process of topological feature transformation is shown in Algorithm 1[1].

3.3 Discriminator Network

The overall network structure of the discriminator is shown in Figure 3.

After the image is processed by the persistent homology module and fed to topological feature transformation, the vector representation (ν_{topo}) of the topological features is obtained. In addition, we transform the features extracted by the original convolutional neural network into vector ν_{conv} for representation and then concatenate these two vectors (ν_{topo}, ν_{conv}) to form a vector (ν).

Here, we process the vector ν using two different network structures: one using a fully connected layer network and the other using a self-attention network.

[1]We use the Python module Gudhi to produce the persistent diagrams.

Algorithm 1: The algorithm of topological feature transformation.

Input: image X of size $H*W$ with C channels.

Output: vector ν_{topo} represents the topological features of image.

1 For X using persistent homology, obtaining a 0-dimensional persistent diagram and a 1-dimensional persistent diagram.

2 Obtaining persistent time τ_k^i using Eq. (3) for each topological feature in 0-dimension and 1-dimension.

3 Obtaining $\nu_{topo} = (\tau_0^1, \tau_0^2, \tau_0^3, ..., \tau_0^n, \tau_1^1, \tau_1^2, \tau_1^3, ..., \tau_1^m)$.

4 Return ν_{topo}.

3.3.1 Fully Connected Layer

We use the network structure of the fully connected layer to process the input concatenated vector ν. The fully connected layer can combine topological features with convolutional neural network features to discriminate between real and fake images. The fully connected layer will learn the most appropriate parameter relationship between these two features during the training process and coordinate the influence of topological features and convolutional neural network features on the discrimination result.

3.3.2 Self-Attention Network

Different from the use of the fully connected layer network structure, the use of the self-attention(Vaswani et al., 2017) network will learn the correlation between the convolutional neural network features and the topological features during the training process, and then discriminate the authenticity of the image. We input the vector ν into the self-attention network, then obtain the vector ν_{sa} after the convolutional neural network features interacts with the topological features. we use the residual network to add the vector ν_{sa} to the original vector ν, as shown in Eq. (4), to obtain the vector ν'. Finally, the vector ν' input to the fully connected layer to judge the authenticity of the image.

$$\nu' = \gamma * \nu_{sa} + \nu \tag{4}$$

Where γ denotes a learnable parameter.

3.4 Loss Function

After we incorporate topological features into GAN, in addition to the original convolutional neural network-based loss, a new topological feature loss

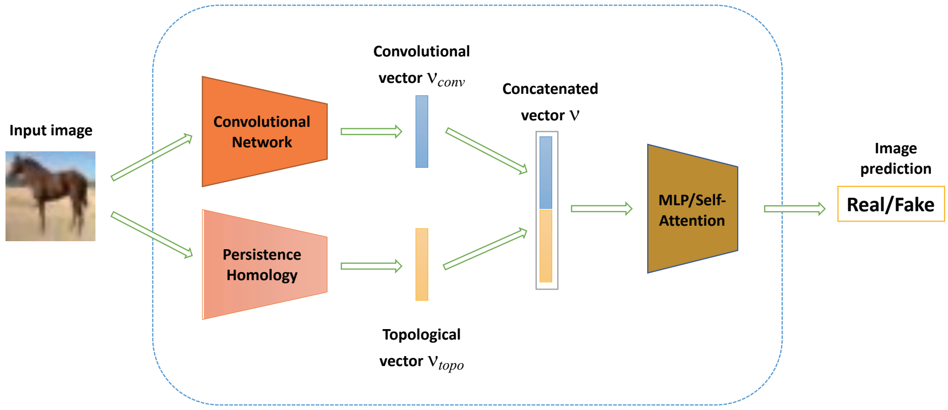

Figure 3: Schematic diagram of the structure of the PHGAN discriminator based on the fully connected layer and the self-attention network.

term is added to the discriminator. In the PHGAN, the generator and the discriminator are alternately trained. When training the discriminator, the topological feature loss term will guide the discriminator to discriminate between real image and generated image in terms of global structure, and then when training the generator, the discriminator can guide the generator to generate an image that is more similar in global structure to the real image. The total loss function of the discriminator and the generator are shown in the following Eqs. (5) and (6):

$$\begin{aligned} &argmaxD[E_{x\sim P_{data}} log(D_{conv}(x) \oplus D_{topo}(x)) + \\ &E_{z\sim P_z} log(1-(D_{conv}(G(z)) \oplus D_{topo}(G(z))))] \end{aligned} \tag{5}$$

$$argmaxG[E_{z\sim P_z} log(D_{conv}(G(z)) \oplus D_{topo}(G(z)))] \tag{6}$$

Where D_{topo} represents discrimination based on topological features. D_{conv} represents discrimination based on convolutional neural network features. $\oplus$ represents the combination of convolutional neural network features and topological features through the fully connected layer and self-attention network structure for discrimination.

See Algorithm 2 for the training process of the PHGAN.

Algorithm 2: The algorithm of training PHGAN.

Input: image X of size $H * W$ with C channels.
Input: *epoch*: number of training iterations.
1 **for** *epoch* **do**
2 Obtaining noise z by randomly sampling.
3 Generating fake image Y using noise z.
4 For fake image Y run steps 7-10, obtaining the discriminant result.
5 For real image X run steps 7-10, obtaining the discriminant result.
6 Update generator and discriminator parameters using Eqs. (5) and (6).
7 For input image run algorithm 1, obtaining topological features vector representation ν_{topo}.
8 For input image, obtaining convolutional features vector representation ν_{conv} by convolutional neural network in discriminator.
9 Obtaining image feature vector representation ν by concatenating ν_{topo} and ν_{conv}.
10 Input ν into MLP/Self-Attention to get the discriminant result.

4 EXPERIMENT

4.1 Experimental Environment and Preparation

We use the CIFAR10 dataset(Krizhevsky, 2012) and the STL10(Coates et al., 2011) dataset for experimental evaluation and comparative analysis with DC-GAN(Radford et al., 2015) , WGAN-GP(Gulrajani et al., 2017) , and WGAN(Arjovsky et al., 2017) .

The CIFAR10 dataset consists of 10 categories of 32x32 color images. Each category contains 6000 images, of which 5000 images are used as training sets and 1000 images are used as test sets. The STL10 dataset consists of 10 categories of 96x96 color images, each with 1300 images, 500 for training, and 800 for testing. In the experiment of this paper, the original image is first cropped into a 32x32 size image by center cropping, and then the training set is used

Table 1: Experimental results on the CIFAR10 dataset.

	DCGAN	WGAN-GP	WGAN	PHGAN$_{mlp}$	PHGAN$_{sa}$
IS↑	5.01	5.05	4.43	5.24	**5.37**
FID↓	66.61	65.00	70.02	64.57	**62.50**
GS(10^{-4}) ↓	10.20	14.50	20.04	9.48	**6.97**

Figure 4: Experimentally generated images base on CIFAR10 dataset.

to train the generative model. In addition to using the FID (Fréchet Inception Distance)(Heusel et al., 2017) and IS (Inception Score)(Salimans et al., 2016) to evaluate the generative model performance, we also used the GS (Geometry Score)(Khrulkov and Oseledets, 2018) evaluation index: a generative adversarial network model generation performance evaluation based on the similarity of topological features.

Experiments are conducted on a Linux server, Ubuntu 18.04 system, and Nvidia Tesla P40 24 GB single graphics card. The Adam optimizer(Kingma and Ba, 2014) with β_1=0.5, β_2=0.999 was used, the batch size was set to 64, and the learning rate during training was 0.0002.

4.2 Experimental Results and Analysis

Table 1 shows the results of our proposed PHGAN on the three image generation metrics of FID, IS, and GS on the CIFAR10 dataset, and compares it with three classic generative adversarial network models: DCGAN, WGAN, WGAN-GP.

It can be seen from the table that the generation results of our proposed PHGAN outperform that of the three comparative GANs on the evaluation indicators of FID and IS. Among them, the PHGAN using the self-attention network (PHGAN$_{sa}$) has better FID and IS evaluation indicators than the PHGAN using the fully connected layer (PHGAN$_{mlp}$), so its experimental performance is the best among the five GANs. From the experimental results, it can be seen that the integration of topological features into the generative adversarial network model can enhance the image generation performance.

In addition, we also use the GS evaluation index to evaluate the experimental results. The GS evaluation index is based on the similarity of the topological features of the generated images and the real images. From the experimental results, we can see that when we use the topological features of the real images to guide the generative adversarial network model, the generated images can better learn the topological feature distribution of the real images. Figure 4 shows the images generated by the experimental five generative adversarial network models on the CIFAR10 dataset. We observe that the images generated by the PHGAN have a clearer overall structure so that it is easier to see the category of the images.

Table 2 shows the experimental results on the STL10 dataset. Similarly, on the FID and IS evaluation indicators, PHGAN performs the best. Different from the experimental results on the CIFAR10

Table 2: Experimental results on the STL10 dataset.

	DCGAN	WGAN-GP	WGAN	PHGAN$_{mlp}$	PHGAN$_{sa}$
IS↑	2.97	2.67	2.81	3.04	**3.12**
FID↓	74.14	79.41	73.96	**71.07**	72.84
GS(10^{-4}) ↓	17.19	39.95	22.32	14.47	**11.68**

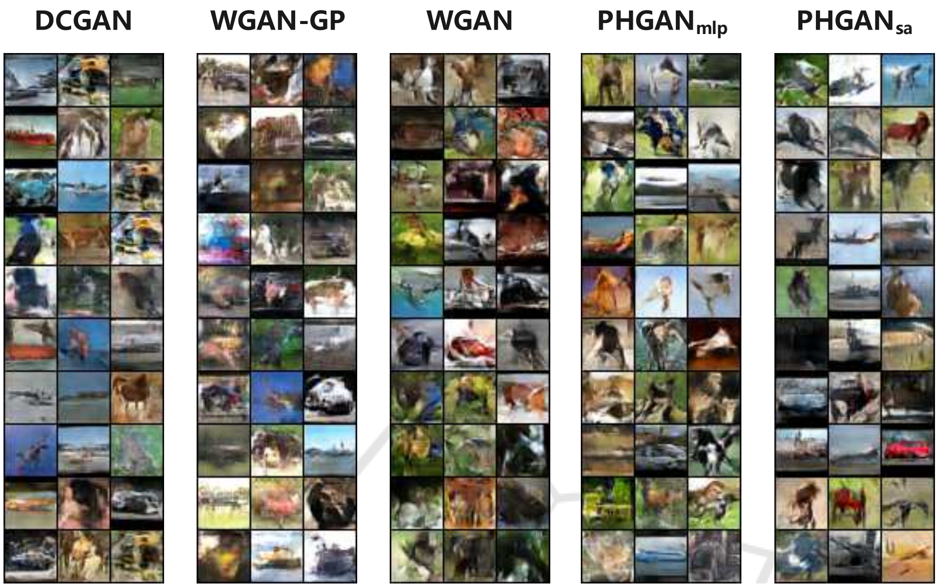

Figure 5: Experimentally generated images base on STL10 dataset.

dataset, the PHGAN$_{mlp}$ is slightly better than the PHGAN$_{sa}$ in the FID evaluation index. It might because the STL10 dataset is relatively small. If we use the self-attention network to process the images, there may be a slight overfitting phenomenon, which leads to the FID indicator not as good as the PHGAN that directly uses the fully connected layer.

Similarly, on the GS indicator, we can also see that PHGAN can learn the topological feature distribution of the real images relatively well on this dataset. Figure 5 shows the images generated by the experimental five generative adversarial network models on the STL10 dataset. We can see that the images generated by PHGAN have sharper boundaries and overall structure.

5 CONCLUSION

This paper proposes the PHGAN which integrates the topological features obtained by persistent homology into the generative adversarial network model. The PHGAN has a good ability to capture global information. It has been verified in experiments. Compared with the original several classic generative adversarial network models, PHGAN has achieved better results in the evaluation matrics for image generation, and the generated images are more realistic. This paper explores the application of the persistent homology method in image generation. And the application in other fields, such as image editing, and image style transfer, is the direction that can be studied in the future.

ACKNOWLEDGEMENTS

This work was supported in part by the Strategic Research and Consulting Project of the Chinese Academy of Engineering under grant 2022-XY-107 and in part by the Shanghai Science and Technology Innovation Action Plan Project under Grant 22511100700.

REFERENCES

Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., and Ziegelmeier, L. (2017). Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

Byrne, N., Clough, J. R., Valverde, I., Montana, G., and King, A. P. (2022). A persistent homology-based topological loss for cnn-based multi-class segmentation of cmr. *IEEE Transactions on Medical Imaging*.

Cang, Z., Mu, L., and Wei, G.-W. (2018). Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLoS computational biology*, 14(1):e1005929.

Carrière, M., Chazal, F., Ike, Y., Lacombe, T., Royer, M., and Umeda, Y. (2020). Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *International Conference on Artificial Intelligence and Statistics*, pages 2786–2796. PMLR.

Coates, A., Ng, A., and Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings.

Gabrielsson, R. B., Nelson, B. J., Dwaraknath, A., and Skraba, P. (2020). A topology layer for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1553–1563. PMLR.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. *Advances in neural information processing systems*, 27.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Hofer, C., Kwitt, R., Niethammer, M., and Uhl, A. (2017). Deep learning with topological signatures. *Advances in neural information processing systems*, 30.

Horak, D., Yu, S., and Salimi-Khorshidi, G. (2021). Topology distance: A topology-based approach for evaluating generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7721–7728.

Khramtsova, E., Zuccon, G., Wang, X., and Baktashmotlagh, M. (2022). Rethinking persistent homology for visual recognition. *arXiv preprint arXiv:2207.04220*.

Khrulkov, V. and Oseledets, I. (2018). Geometry score: A method for comparing generative adversarial networks. In *International conference on machine learning*, pages 2621–2629. PMLR.

Kim, K., Kim, J., Zaheer, M., Kim, J., Chazal, F., and Wasserman, L. (2020). Pllay: Efficient topological layer based on persistent landscapes. *Advances in Neural Information Processing Systems*, 33:15965–15977.

Kindelan, R., Frías, J., Cerda, M., and Hitschfeld, N. (2021). Classification based on topological data analysis. *arXiv preprint arXiv:2102.03709*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A. (2012). Learning multiple layers of features from tiny images. university of toronto (2012). *URL: http://www.cs.toronto.edu/kriz/cifar.html, last accessed*, 5:13.

Li, Y., Xuan, Y., and Zhao, Q. (2022). Manifold projection and persistent homology. *Measurement*, page 111414.

Merelli, E., Rucco, M., Sloot, P., and Tesei, L. (2015). Topological characterization of complex systems: Using persistent entropy. *Entropy*, 17(10):6872–6892.

Mileyko, Y., Mukherjee, S., and Harer, J. (2011). Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12):124007.

Moor, M., Horn, M., Rieck, B., and Borgwardt, K. (2020). Topological autoencoders. In *International conference on machine learning*, pages 7045–7054. PMLR.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, 29.

Schiff, Y., Chenthamarakshan, V., Hoffman, S. C., Ramamurthy, K. N., and Das, P. (2022). Augmenting molecular deep generative models with topological data analysis representations. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3783–3787. IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Z., Ren, Q., Wang, J., Yan, C., and Jiang, C. (2022). Mush: Multi-scale hierarchical feature extraction for semantic image synthesis. In *Proceedings of the Asian Conference on Computer Vision*, pages 4126–4142.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2019). Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Ziou, D. and Allili, M. (2002). Generating cubical complexes from image data and computation of the euler number. *Pattern Recognition*, 35(12):2833–2839.

Zomorodian, A. and Carlsson, G. (2004). Computing persistent homology. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 347–356.