Turkish Journal of Electrical Engineering and Computer Sciences

Volume 31 | Number 2 Article 10

1-1-2023

3D Point Cloud Classification with ACGAN-3D and VACWGAN-GP

ONUR ERGÜN

YUSUF SAHİLLİOĞLU

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2023) 31: 381 – 395
© TÜBİTAK
doi:10.55730/1300-0632.3990

3D point cloud classification with ACGAN-3D and VACWGAN-GP

Onur ERGÜN*, **Yusuf SAHİLLİOĞLU**
Department of Computer Engineering, Middle East Technical University, Turkey

Received: 14.08.2022 • **Accepted/Published Online:** 01.02.2023 • **Final Version:** 23.03.2023

Abstract: Machine learning and deep learning techniques are widely used to make sense of 3D point cloud data which became ubiquitous and important due to the recent advances in 3D scanning technologies and other sensors. In this work, we propose two networks to predict the class of the input 3D point cloud: 3D Auxiliary Classifier Generative Adversarial Network (ACGAN-3D) and Versatile Auxiliary Conditional Wasserstein Generative Adversarial Network with Gradient Penalty (VACWGAN-GP). Unlike other classifiers, we are able to enlarge the limited data set with the data produced by generative models. We consequently aim to increase the success of the model by training it with more data.

As suggested by the conventional ACGAN models, in addition to the real dataset, we train the Discriminator with synthetic data generated by the Generator using the class label. By doing so, we ensure that Discriminator can discriminate between the real data and the synthetic data. Thus, as the training evolves, the Generator is trained to produce more realistic synthetic data, which in turn forces Discriminator to classify or discriminate better. Defined originally on 2D images, our ACGAN-3D modifies this conventional ACGAN model in order to classify 3D point clouds by updating the neural network layers.

Our second model VACWGAN-GP, on the other hand, demonstrates similar abilities with more stable training by replacing its Discriminator with Critic and by modifying its loss function. In this model, we managed to merge Wasserstein GAN-GP with conditional GAN in order to improve the classifier's performance.

The proposed models ACGAN-3D and VACWGAN-GP were tested extensively on 3D datasets and comparisons with the other state-of-the-art studies have revealed our clear advantages on various aspects. While ACGAN-3D can be preferred with its compact design, our second method VACWGAN-GP stands out for higher performance.

Key words: Deep learning, GAN, CGAN, ACGAN, Wasserstein GAN, Wasserstein GAN-GP, 3D object classification

1. Introduction

In simple terms, classification is the task of deciding to label of the given sample with some techniques that, for instance, learn from other known samples. Binary classification problems assign samples to one class or another, e.g., whether an email is spam or not, whereas multi-class classification categorizes samples using one of the many classes. In the multi-class case, one sample can have more than one label, e.g., identification of the categories of a film.

The success of classifying 2D images and videos has paved the way for 3D object classification. In this study, we utilize generative models for the classification task of 3D objects represented as 3D point clouds. This data can come directly from sensors or can be obtained by taking vertex or point samples from 3D digital mesh models. Thanks to the availability of economical data sensors, the usefulness of the 3D modeling software packages, and the simplicity of sharing over the world wide web, this data became very common in important

*Correspondence: onrrgn@gmail.com

applications such as object classification. A self-driving car or a robot, for instance, needs to classify such three-dimensional data in order to make important decisions concerning navigation and grabbing.

In order to increase the classification success, we propose using the data generated by the generator that is close to the real dataset distribution. To this end, we propose both a modified ACGAN for 3D point clouds (ACGAN-3D) and a combined version of a Wasserstein GAN-GP method with a classifier (VACWGAN-GP). For the latter, we adapt the gradient penalty part to be able to work with 3D point cloud data. As a result, we present methods for labeling 3D shapes without using connectivity information that characterizes 3D objects.

Contributions An Auxiliary Classifier Generative Adversarial Network (ACGAN), called ACGAN-3D, is presented as a novel approach to the 3D object classification problem. Our network allows the classification of 3D data using only the position information of the input points. The Discriminator of this network is utilized as a classifier and the dataset has been augmented owing to the data generated by the Generator during training, resulting in an enhanced generalization of the model. The suggested method allows the generator to be multiplied up to the number of target classes theoretically, thus overcoming the possible mode collapse problem during training and enabling each Generator to be used as a data generator for the specific classes after training. Having a more compact design, ACGAN-3D is slightly faster than our second proposal VACWGAN-GP. We also benefit from the generative power of Wasserstein GAN-GP (WGAN-GP) by combining it with a separate classifier in our second independent model. To this effect, we customized the gradient penalty part for our 3D classification purposes.

In general, we utilize the main motivation of using GAN models, i.e. learning the distribution of the real but sparse dataset and undertaking the task of data augmentation automatically. We inherit this feature into novel point cloud classification solutions in order to obtain more robust classifiers.

Our contributions can be summed up as follows:

- ACGAN is used for the first time in the classification of 3D points equipped with only coordinate information. We call this version ACGAN-3D.
- With multiple generator support, we avoid the potential GAN-specific mode collapse problem.
- Inspired by the spirit of the VACGAN approach, we use the generative model of Wasserstein GAN-GP for the first time in 3D point cloud classification.

2. Related work

Although our focus in this paper is on 3D classification, 3D recognition, and segmentation are also closely related problems. We, therefore, investigate the recent learning-based literature on these areas.

In traditional machine learning, descriptors that are relevant to the 3D model's properties are first selected, and then the attributes are acquired. The point cloud is segmented into relevant pieces based on these characteristics. This method's drawback is that it is heavily reliant on descriptors and is unsuitable for complicated data [1]. Furthermore, because 3D descriptors are very highly dimensional, it tends to over-fit [2]. Qi et al. [3] indicate that manually designed feature development or selection is difficult and application-dependent. Learning techniques based on such characteristics are confined to the capacities of those features to represent the original data, not to the original data itself. This imposes an unnecessarily restrictive constraint on learning algorithms. As a remedy, they promote the deep learning machinery in action for 3D classification and segmentation problem.

Wu et al. [4] presented another deep network in the form of a belief net on 3D data. They turned the unstructured input data into a volumetric representation to be utilized as input in their technique, rather than utilizing a view-based approach. A convolutional deep belief network is used in this innovative application to represent a 3D model as a probability distribution of binary variables on a 3D voxel grid. Their method, 3DShapeNets, gets a 2.5D depth picture as input and constructs a volumetric structure in which the shape distribution is learned. Han et al. [5] then introduced a local shape descriptor that combines the benefits of the deep belief networks in unsupervised learning and the advantages of convolutional networks to propose a circle convolutional restricted Boltzmann machine.

VoxelNet of [6] described a method for classifying objects in point clouds. The researchers used point clouds to build voxels, which are then run through an artificial neural network. [7] illustrated how CNN may be used to directly identify a road using LIDAR data as input. Hybrid systems such as multi-view CNNs [2], transform 3D point clouds into 2D pictures and train a neural network to recognize things in those images. Although their system performs better than ShapeNets [4] in retrieval applications, one disadvantage of this strategy is that it eliminates the benefits of using 3D data in the first place. This includes tasks like point categorization, scene completion [8], and scene comprehension [9], all of which can help an automobile comprehend its environment.

Training instability problem of generative adversarial networks (GAN) is alleviated by the introduction of Wasserstein GAN, which is made more robust by replacing the clipping weights with a penalty term, yielding the WGAN-GP [10]. Auxiliary classifier GAN (ACGAN) [11], on the other hand, turns out to be a robust architecture that can generate globally coherent ImageNet samples. We lift both WGAN-GP and ACGAN from 2D to 3D for our classification purposes.

Our task of 3D object classification directly from point clouds has been comprehensively investigated in a recent survey [13]. In PointNet [3], researchers trained a neural network to recognize an item based on the raw data points gathered. PointNet takes input samples from a point cloud and outputs class labels for each point in the input for point-wise classification (or semantic segmentation). The PointNet study also presents analogous designs for object classification and object-part segmentation, in addition to point-wise classification. Qi et al. improve the PointNet approach with PointNet++ [14] taking into consideration local features. They proposed a hierarchical structure that employs PointNet recursively on nested parts of the input. Most recently, Song et al. [15] employed multiple 2D convolutional layers for coordinates of points and applied max pooling layer like PointNet in order to build global features for shape. Proposed Pointwise CNN was subject to classification task on Modelnet10 dataset during their experiments. Anguelov et al. [16] proposed an octree grouping-based network structure for PointNet++, named OG-PointNet++, which calculates the point density by creating an imbalanced octree for the point cloud. It then groups points based on the calculated density. These point groups are allocated to different layers based on their density, then PointNet++ extracts the local characteristic of each group. The last abstract layer yields the global feature, which is utilized for classification and segmentation. Experiments demonstrate that it performs well in a variety of 3D tasks, including object categorization and semantic segmentation. In [12], conditional random fields are used to classify 3D human limbs in point clouds.

In VoxNet [17], Maturana et al. proposed a method integrating volumetric occupancy grid representation with a supervised 3D CNN in order to cope with large amounts of point cloud data. They trained their proposed network on LIDAR point-cloud data, RGB-D and modelnet datasets. Thus, they support different sources in the manner of 3D data production. Shi et al. [18], on the other hand, converts 3D shapes into a panoramic view with respect to their principal axes in DeepPano in order to solve the 3D shape recognition problem.

Thus, suggested deep CNN becomes invariant to the rotation around the principle axis for 3D shape queried. They examined their methods on the Modelnet10 and Modelnet40 datasets. Kumawat et al. propose a new block as an alternative to 3D convolutional networks in their LP-3DCNN work [19] in order to increase the learning capacity of CNN networks and to overcome the problems caused by CNN networks such as large model size, computational and space complexity etc. This block aims to extract feature maps by evaluating 3D local neighborhoods. While testing their proposed method on volumetric data, they use voxels in their networks and conducted experiments with their method on classification task of modelnet10 and modelnet40 datasets. Khan et al. developed a pipeline for generative model in order to generate 3D Shapes and that pipeline consists of sequential transformations from coarse to fine [20]. Their GAN method employs primitive parts as features and improves itself by adding fine scale details. As a way of demonstrating the learning power of their proposed generative model, they chose to run experiments using it as a feature in the classification task. Recent trend in object detection and classification shifts towards the Transformer network architecture [22] that is based solely on attention mechanism. To this end, Misra et al. [21], Yuan et al. [24], and Lu et al. [23] showed remarkable success in leveraging Transformers to learn 3D object detectors and classifiers without relying on hand-designed inductive biases. A regularization strategy that is agnostic to the network architecture [25] relies on one-to-one correspondences [27] between two point clouds during classification. [28], on the other hand, uses rendered 2D images of 3D point clouds from random camera viewpoints in a self-supervised fashion to classify its input. We compare our work with most of these studies as well as the PointNet and ShapeNet family and demonstrate advantages on various aspects.

3. Methodology

In this work, inspired by 2D classifiers based on GAN models, we propose two different approaches for classification on 3D point clouds and then compare our results with the well-known study, PointNet [3]. The first method we propose, called ACGAN-3D, includes an adapted version of the ACGAN model designed for 2D images. We adapt it for 3D point cloud classification. We rearrange the network layers of the Discriminator and Generator models for the adaptation. In the network structure, we keep the original ACGAN topology and set the weights W_{adv} and W_{ac} in the discriminator loss function (Eq. 1) equal to each other. The architectural structure of the proposed ACGAN-3D is given in Figures 1-a.

$$\begin{aligned} L_{adv} &= E_x[\log D(x)] + E_{\hat{x}}[\log(1 - D(\hat{x}))] \\ L_{ac} &= E_x[\log p(c|x))] + E_{\hat{x}}[\log(1 - p(c|\hat{x}))] \\ L_D &= \frac{W_{adv} * L_{adv} + W_{ac} * L_{ac}}{W_{adv} + W_{ac}} \end{aligned} \tag{1}$$

We also propose another method, called VACWGAN-GP, to perform the classification task based on the VACGAN method. To this end, we replace the vanilla GAN part of VACGAN with Wasserstein GAN (WGAN) that uses gradient penalty. Different from the general structure of WGAN, Eq. 2 defines the loss for generator and critic, respectively. The cost function of WGAN is defined in Eq. 3 where f is 1-Lipschitz function.

$$Loss_G = \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log{(D(G(z^{(i)})}] \tag{2}$$

$$Loss_C = \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log{(D(G(z^{(i)})}$$

$$\min_{G} \max_{||f||_L \leq 1} E[f(x)] - E[f(\hat{x})] \tag{3}$$

Our network layer structure is being updated as shown in Figure 1-b so that our VACWGAN-GP technique can work properly with the 3D point cloud input.

The layers of classifier parts of both networks we propose have a similar approach to the PointNet study with some hyperparameter changes.

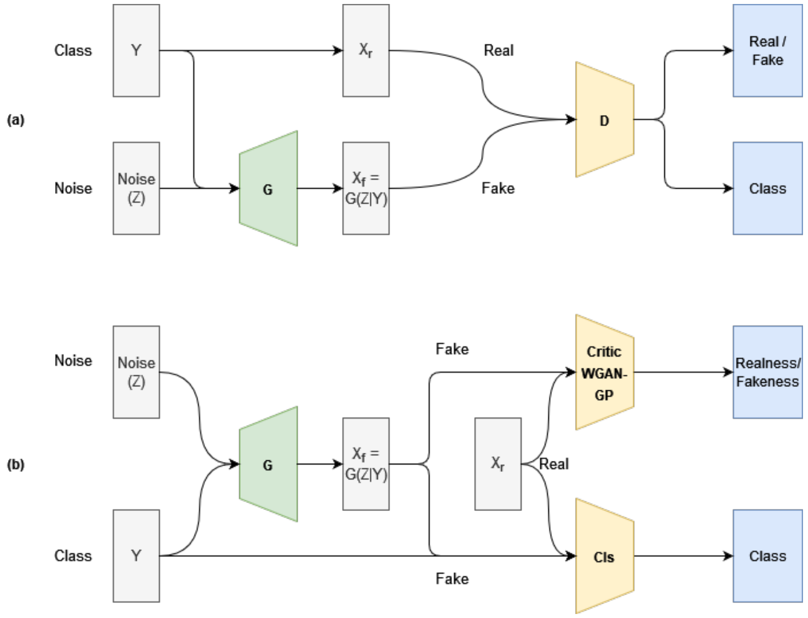


Figure 1. Proposed models: (a) ACGAN-3D and (b) VACWGAN-GP for 3D point cloud classification.

In GAN models, the Z input is collected from the latent space to feed the Generator and has no real meaning on its own. Using this input, the Generator tries to generate meaningful data as close to real data as possible. If the generated data can be produced well enough, which means representing the real dataset well, it undertakes the task of data augmentation. If the relationship between the data is acceptable enough, training through viewing more data produces higher outcomes in terms of classification success. Therefore, the classifier will generalize the data better and give more outstanding results. We propose to apply generative models for data augmentation to increase our classification success based on this claim. For this purpose, we compare the results of the two methods we proposed with a bare classifier that reflects PointNet classification spirit.

Similar to other GAN models, it is possible to encounter a mode collapse problem in ACGAN models. If Discriminator can no longer update itself and gets stuck on local minima, for example, the Generator can trick this Discriminator with generating a small dataset as an output. Generator restricts itself to generate the data set similar with less variation.We integrate Wassterstein GAN-GP with VACGAN to avoid a potential mode collapse problem and provide more stable training.

One of our main goals is to expand the data set for the classifier with generated fake data that can best reflect the distribution of the original data set. In this regard, WGAN with GP is ideal as it promises to produce the closest data set to the input with the help of a defined gradient penalty. However, in the original work, this gradient penalty is described for 2D images. In our study, we modify the distance measure defined in

WGAN-GP with Chamfer Distance to fit measuring the distance between 3D point clouds. Thus, for the critic, we update the gradient penalty part in the loss objective defined in Eq. 4 to use Chamfer Distance. Chamfer Distance metric is continuous and can be expressed as in Eq. 5, where S_1 and S_2 are point sets defined in $\mathbb{R}^3$.

$$L = \underset{\hat{x}\sim P_g}{E}[f(\hat{x})] - \underbrace{\underset{x\sim P_r}{E}[f(x)]}_{\text{critic loss}} + \lambda \underbrace{\underset{\hat{x}\sim P_{\hat{x}}}{E}[(||\nabla_{\hat{x}} f(\hat{x})||_2 - 1)^2]}_{\text{gradient penalty}} \tag{4}$$

$$d_{CD}(S_1, S_2) = \sum_{x\in S_1} \min_{y\in S_2} ||x-y||_2^2 + \sum_{y\in S_2} \min_{x\in S_1} ||x-y||_2^2 \tag{5}$$

The method we propose is similar in spirit to the multiple objective generative adversarial active learning [29] method proposed by Yezheng Liu et al. While their work aims to use Generative models to better define the boundaries of normal data in outlier detection, in our method, the generative model learns the distribution of the data set and produces data close to the real data, thus allowing the classifier to generalize more effectively.

3.1. Implementation and test details

The structure of both Generator and Discriminator models we used during our experiments to test and evaluate our first method, ACGAN-3D, are given in Figure 2-(a), where L is latent vector dimension, n is point set size, and k is target class size.

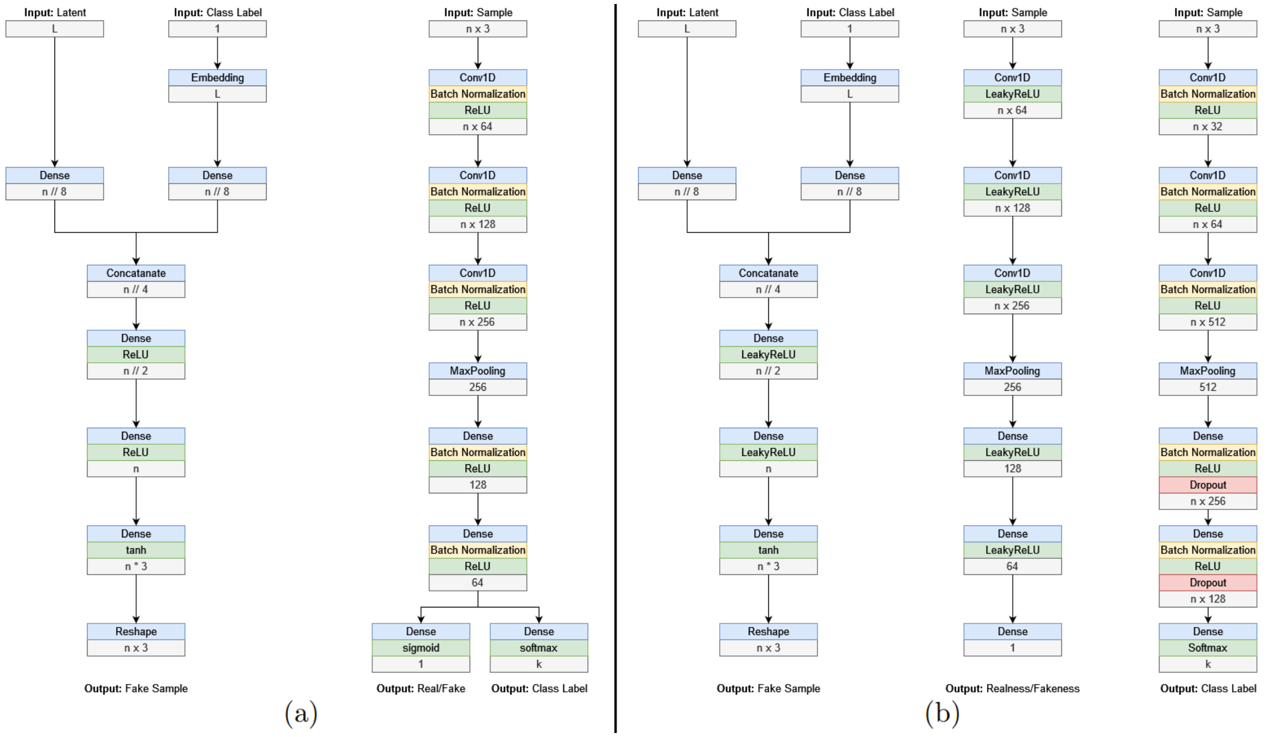


Figure 2. (a) Proposed ACGAN-3D Structure with layers. Left: Generator, Right: Discriminator/Classifier. (b) Proposed VACWGAN-GP Network Structure with layers. Left: Generator, Middle: Critic, Right: Classifier. L: Latent vector dimension, n: Point set size, k: Target class size.

The layer structures of the models of our second and promoted method VACWGAN-GP are given in Figure 2-(b). Note that, batch normalization layers are omitted in the critic since the correlation created between the samples in the same batch by those layers has a bad impact on gradient penalty effectiveness.

Sample point cloud size is given as n and k is the number of labels for classes and latent vector dimension is expressed with L for both models. In our ACGAN-3D model, Adam is employed as a gradient optimizer in both Discriminator and Generator, whereas in the proposed VACWGAN-GP model, the RMSprop optimizer is selected for Generator and Critic. Adam optimizer is used by the classifier.

In order to increase the generator performance and indirectly the discriminator and the critic performance, we take this Z input from a Gaussian distribution space with mean 0 variance 1 as suggested in [30]. In addition, during the preprocessing step of training, we scale the input data between -1 and 1 such that each point vector relies on a defined boundary cube without no distortion. As a result, we now have a model that is more stable and that can converge quickly.

4. Experiments

In this section, we describe our experiments with these two proposed methods (ACGAN-3D and VACWGAN-GP) and discuss their results. We evaluate and compare the training outcomes of our GAN-based proposals. We also show whether a sample point set size or latent vector size change has an impact on model performance. In addition to the analysis of these methods for the shape classification task given in the sequel, we also observe plausible shape retrieval results. To implement the retrieval task, a sample from each Modelnet10 class is randomly selected from the test set as query data and then similar shapes are retrieved from the dataset. Using this selected query data, similar shapes are retrieved from the data set in a protocol similar to [31, 32].

We tested the sufficiency of the proposed models over the 3D point clouds of Modelnet10 and Modelnet40 datasets. While developing our methods, we compare them with our base model as well as other well-known approaches such as PointNet [3] and PointNet++ [14]. We compare the results of our tests with the other methods that work on the same dataset. Unless otherwise noted, all tests were carried out on the Modelnet10 dataset since the desired data class set is small and hence it is easy to explain and depict the process. Furthermore, the comparisons will be significant due to the widespread use of this dataset in other well-known studies. In addition, the point sample set size n is chosen as 2048 per each mesh in the dataset during the experiments. We perform uniform sampling that picks surface mesh triangles with probabilities proportional to their areas and then puts a uniformly distributed random sample inside the pick. We prefer this method over an artificial data generator alternative such as [26] in order to be compatible with the other studies for comparisons.

4.1. Datasets

4.1.1. Modelnet10

The Modelnet10 dataset contains CAD models created from the most widely used object categories in the world. The samples are manually categorized. In this data provided by Princeton University [1], CAD models' orientation are manually aligned. It is a subset of the Modelnet40 dataset and has 10 different object categories: Bathrub, Bed, Chair, Desk, Dresser, Monitor, Night Stand, Sofa, Table, and Toilet. In the dataset, the train and test set are separated and this split is employed exactly as specified during the experiments.

[1] https://modelnet.cs.princeton.edu/

4.1.2. Modelnet40

Modelnet40 dataset is a superset of the Modelnet10 dataset with 30 different new categories. Similar to Modelnet10, this dataset is also divided into train and test, and this seperation is used during our experiments without any changes. Number of training models is 3991 and 9843 for Modelnet10 and Modelnet40, respectively. Similarly, 908 test models on Modelnet10 is expanded into 2468 models in Modelnet40.

5. Results

In this section, we share the results of the proposed modified ACGAN-3D and VACWGAN-GP methods during our experiments. The results contain the proposed GAN models' learning-related loss charts. In addition, we provide the results that we use to assess categorization success. We also show examples of mislabeled samples. Furthermore, we discuss the impact of modifying the size of the latent vector or the number of sampled points on classification performance for both methods.

5.1. ACGAN-3D

Our ACGAN-3D consists of one generator and one discriminator network where the discriminator is also responsible for the classification task. Figure 3-left shows how both generator and discriminator losses propagate through the iterations of training. As expected, spikes with large variance formed in the early stages of training reach equilibrium by shortening towards the end of training. This shows us that there is healthy GAN training. As can be observed from Figure 3-right that plots the change in classification loss through epochs, the loss value on fake data starts higher than the loss value in the real data and decreases more with a higher acceleration compared to the real data. Although the equilibrium period takes a little longer for fake data, it is seen that the equilibrium level is reached for both types. This shows us that fake data converge into real data and generator increases classification success by producing data close to real data. The change in classification accuracy value on both the train set and the test set during the epochs of the training can be seen in Figure 4-(a) along with the classification losses in Figure 4-(b). The consistent improvement in accuracy and lack of sharp spikes in both the train and the test set during training support our statements above.

When we check the classification success of our ACGAN-3D method via its confusion matrix, it is seen that especially the two classes, namely the night stand and dresser, are mixed with each other and most of the misclassified samples belong to these two classes. When the objects of these two classes are examined visually, it is seen that the night stand and dresser converge to a common cube shape geometrically (Figure 5 top row corner). More of our classification results in comparison with three point-based methods from Table 1 can be found at Figure 5. In addition to the chair object, we demonstrate the objects that performed the worst (night stand) and best (bed) in our evaluations. In this figure, the actual class labels and the estimated class labels are given with probabilities that define belonging to the associated class. The corresponding confusion matrix is provided in Figure 6.

5.2. VACWGAN-GP

In the second proposed method, VACWGAN with GP (VACWGAN-GP), the GAN part of the existing VACGAN method was replaced with Wasserstein GAN with gradient penalty, and unlike ACGAN, the classifier network is kept separate from the GAN model. With the use of the Chamfer Distance function in the calculation of the gradient penalty, the generator model produces fake data closer to the distribution in the real dataset and it takes on the task of data augmentation that feeds the classifier with more data. Therefore, improved

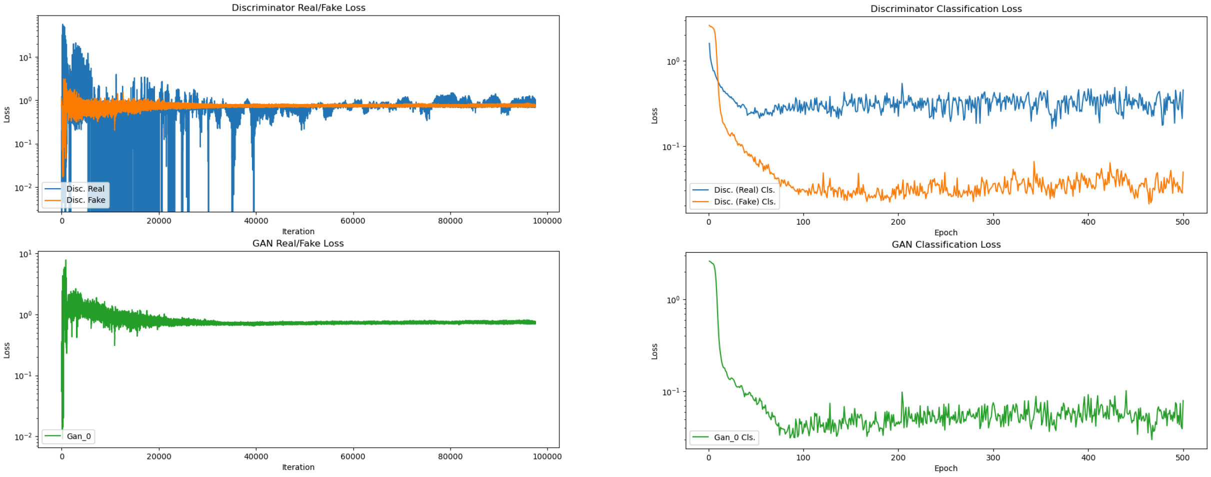
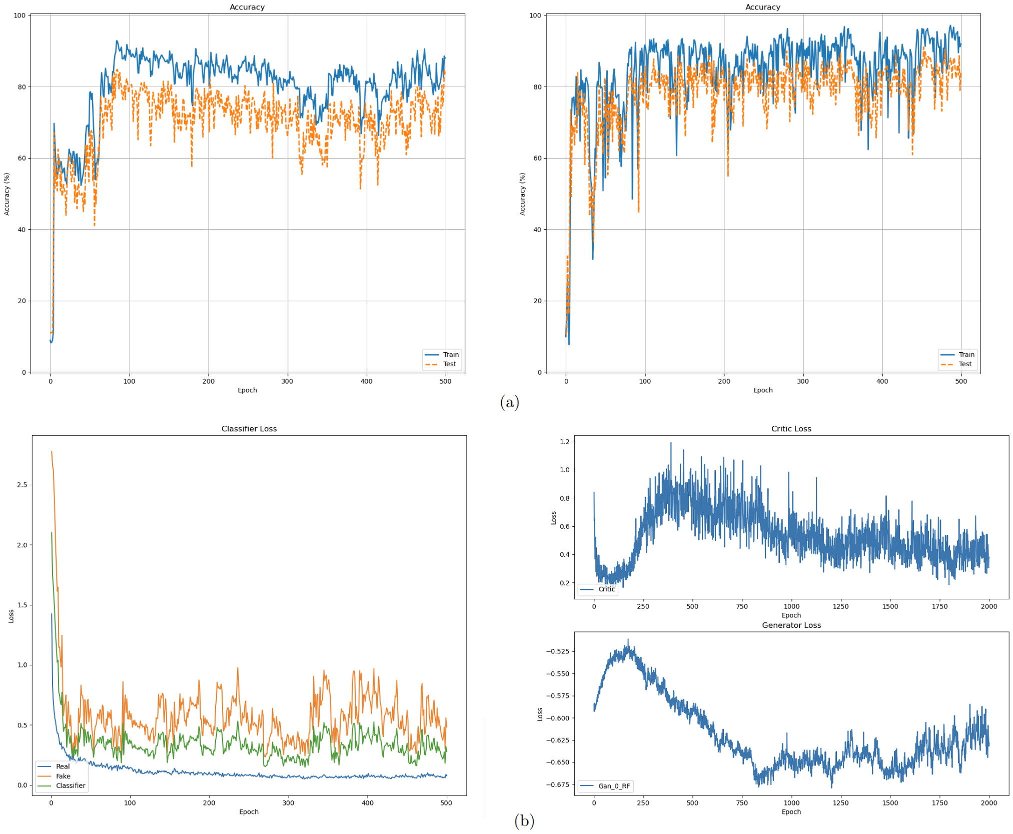


Figure 3. (Left) Discriminator (top) and generator (bottom) real/fake losses of our ACGAN-3D through iterations. (Right) Discriminator (top) and generator (bottom) classification losses of our ACGAN-3D through epochs.



(a)

(b)

Figure 4. (a) Classification accuracy of our ACGAN-3D (left) and our VACWGAN-GP (right) with different epochs. (b) (Left) Classification loss of our VACWGAN-GP through epochs. (Right) Critic (top) and generator (bottom) loss of our VACWGAN-GP through epochs.

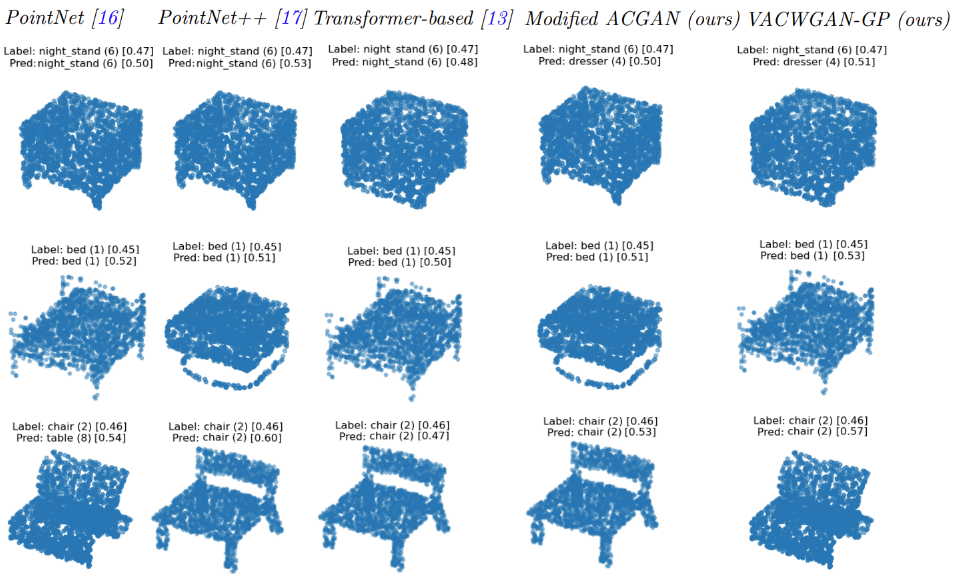


Figure 5. Classification results of our methods (columns 4 and 5) and competitors (columns 1-3) for three different queries. Label row for each sample indicates the real label of the sample and the value in brackets shows the assigned probability for that class. Pred row for each sample, on the other hand, expresses the label assigned by classifier for that sample with the probability of to belonging that class. Samples are from Modelnet40 and they are represented by 2048 points.

Table 1. Classification values for the proposed methods. F1-s. is short for F1-score and Sup. is for Support.

Modelnet10	**ACGAN-3D**			**VACWGAN-GP**			**Sup.**
	Prec.	Rec.	F1-s.	Prec.	Rec.	F1-s.	
Bathtub (0)	0.97	0.78	0.87	0.98	0.84	0.90	**50**
Bed (1)	0.90	0.98	0.94	0.91	0.98	0.94	**100**
Chair (2)	0.94	0.89	0.91	0.96	0.98	0.97	**100**
Desk (3)	0.66	0.66	0.66	0.89	0.74	0.81	**86**
Dresser (4)	0.58	0.79	0.67	0.81	0.84	0.82	**86**
Monitor (5)	0.84	0.97	0.90	0.93	0.99	0.96	**100**
Night Stand (6)	0.88	0.42	0.57	0.82	0.77	0.80	**86**
Sofa (7)	0.93	0.97	0.95	0.97	0.97	0.97	**100**
Table (8)	0.82	0.84	0.83	0.86	0.96	0.91	**100**
Toilet (9)	0.96	0.93	0.94	0.99	0.95	0.97	**100**
Accuracy			**0.83**			**0.91**	**90.8**
Macro Avg.	0.85	0.82	0.82	0.91	0.90	0.91	**90.8**
Weighted Avg.	0.85	0.83	0.83	0.91	0.91	0.91	**90.8**

classification results are achieved. Similar to Section 5.1, we investigate losses, accuracies, and confusions. It is seen that as the epochs in the training progress, the stability in the loss value for real and fake data increases and decreases in parallel, which is already observed for our ACGAN-3D. Due to the WGAN nature, networks establish equilibrium in a slower but more stable manner. The classification accuracy has also increased

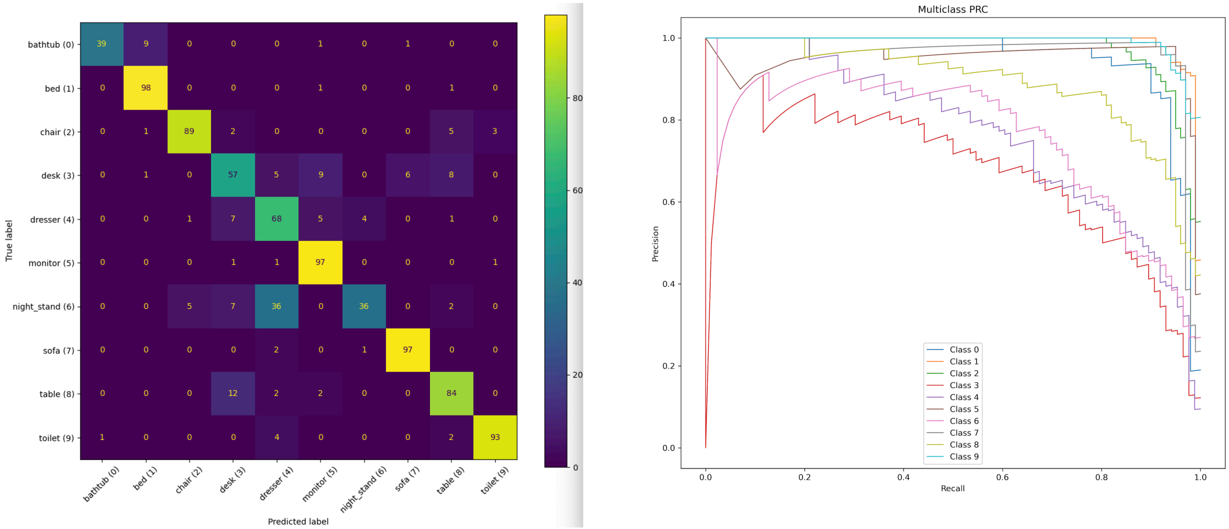


Figure 6. Confusion matrix (left) and precision-recall curve (right) of our ACGAN-3D results.

decisively. There is no doubt that the success of classification increases compared to the other two base methods: PointNet and our ACGAN-3D. We would like to draw attention to the probabilities given to the correct classes for some of the incorrectly guessed examples. Our classifier, which comes close to the correct classification, actually mislabels the data that may be difficult to label even visually.

5.3. Impact of latent vector dimension

During the experiments, we employed some tests that use different sizes for latent vector in order to observe how latent vector dimension affects model performance. Thus, we repeat our tests by only updating the latent vector size to 50, 100, 256, and 512 while leaving everything else the same. We would like to draw attention to the fact that we are considering the generator network structure when choosing the number 256 as the limit. Generator, which takes the class value as an input, embeds this input to be the same size as the latent vector, and combines this layer with the latent vector. We do not want the output of this layer to be larger than the successor layer. As suggested in other studies, choosing a high latent vector size in our experiments also has negative effects such as late convergence or failure to converge at all. It is observed that there is no significant change that will affect the classification results among other values. As a result, 256 is chosen as a latent vector size throughout the tests.

5.4. Impact of point set size

Both of the models we propose can take sampled point set size as a configuration parameter and then work with the specified number. However, like with the latent vector size, it may be helpful to take into consideration the network layer's structure and dimension. In order to demonstrate our flexibility in the manner of sampled point cloud size, we repeat our tests by sampling from the data set at various densities. We chose 128, 256, 512, 1024, and 2048 as candidate sizes. Our tests demonstrate that the success of classification drops by roughly 3% on average as we decrease to the next candidate size. Due to the difficulty of expressing the shapes with details at 128 and fewer samples increases, drastic decreases are observed in the classification accuracy. In order to be able to make a more fair comparison with the other studies, we conducted our tests on the number of 2048 points, taking PointNet as a reference.

5.5. Impact of missing data

It may not always be possible to have the desired number of points for the 3D point clouds to be queried. In particular, it is possible to encounter this situation in the data produced by the sensors. In order to examine the behavior of the proposed methods under missing data and to show their strength, we conduct experiments with different rates of missing data. In this context, the coordinate information of randomly selected points from each sample in the point cloud data generated from the test set in the Modelnet10 data has been cleared. The results of the experiments with different data reduction percentages reveal our generalization capability. The results also show that both methods we propose are resistant to data loss.

5.6. Comparisons and complexities

Commonly used precision, recall, and F1-score values based on the true/false positives/negatives are provided for our two proposals ACGAN-3D and VACWGAN-GP in Table 1, which shows that the two methods we propose perform quite well in classifying the dataset. Our VACWGAN-GP method, in particular, outperforms our ACGAN-3D method and competes with state-of-the-art methods. While ACGAN-3D method is more inaccurate in categorizing samples belonging to Desk, Dresser and Night Stand classes, VACWGAN-GP produces decent results. Table 2 compares the classification accuracy of our methods with previous studies. We observe both of our methods' comparable performances with these nine state-of-the-art methods while we require a lighter training process. Overall, both our methods possess light-weight neural network layers and do not include preprocess layers, unlike the deep and complex alignment layer in our competitors that explains their better performance on Table 2. ACGAN-3D is even lighter than VACWGAN-GP in that it includes its own classifier.

Table 2. Classification accuracy comparison of our methods (last 2 rows) with the competitors. M10 is short for Modelnet10.

Method	**Input**	**Acc. M10**	**Acc. M40**
3DShapeNets [4]	Volume	83.5%	77%
VoxNet [17]	Volume	92%	83%
LP-3DCNN [19]	Volume	94.4%	92.1%
Primitive-GAN [20]	Volume	86.4%	92.2%
DeepPano [18]	2D (Pano)	85.45%	77.63%
PointNet [3]	Point	77.6%	89.2%
PointNet++ [14]	Point	-	90.7%
Transformer-based [21]	Point	-	89.1%
Pointwise CNN [15]	Point	87.07	-
Modified ACGAN (ours)	Point	85.2%	75.4%
VACWGAN-GP (ours)	Point	91.74%	81.3%

We further provide space and time complexity analyses on our light models. The number of the trainable parameters in the networks in ACGAN-3D is 13.99M and 0.49M for Generator and Discriminator+Classifier, respectively. For VACWGAN-GP, 15.34M, 0.08M, and 0.2M parameters are used by Generator, Critic, and Classifier, respectively. Space complexity is $O(N)$ where N is point cloud size of the samples.

We explore the training and testing timing separately. Nvidia GeForce RTX 3090 24GB graphics card is used to train neural networks and it takes between 8 h to a day depending on the epoch desired. Our test time for the classification of an instance is 0.0010 and 0.0013 s with ACGAN-3D and VACWGAN-GP, respectively. Classification time in s on the testset of Modelnet10 and Modelnet40 dataset is 0.99 and 2.53, respectively, for ACGAN-3D, and 1.18 and 2.88 for VACWGAN-GP. As verified by these timings, both our models scale well thanks to their linear $O(N)$ time complexities.

6. Conclusion and future work

In this paper, we propose two different methods to classify 3D shapes represented as plain point clouds without normal information. With the proposed methods ACGAN-3D and VACWGAN-GP, we use the power of generative models to make our classifiers better. Our ACGAN-3D is a modified ACGAN, which is already designed to perform the classification task on 2D images. We made ACGAN-3D suitable for 3D classification by updating network layers. In our second and promoted solution we propose replacing the GAN model part with Wasserstein GAN. On top of the Wasserstein GAN structure, we apply the gradient penalty, which can also solve problems such as vanishing gradients and mode collapse. We specifically replace the conventional gradient penalty calculation available for 2D images with the Chamfer distance in order to suit the nature of the 3D point cloud problem. To the best of our knowledge, we implement such use of Wasserstein-GAN with Chamfer distance gradient penalty as a data augmenter in 3D point cloud classification for the first time in literature. Both methods we propose are acceptably sensitive to the number of points selected, as long as they are compatible with the network layer structure and comply with the physical boundaries. Furthermore, the models we propose are resistant to working with incomplete data, which means they perform better against data sets that may be incompletely generated by sensors. In conclusion, while the first of the two models we propose, ACGAN-3D, can be preferred with its compact design, our second method VACWGAN-GP stands out for higher performance. We also developed a fast and robust shape retrieval application (Figure 7) that employs these two proposed classifiers.

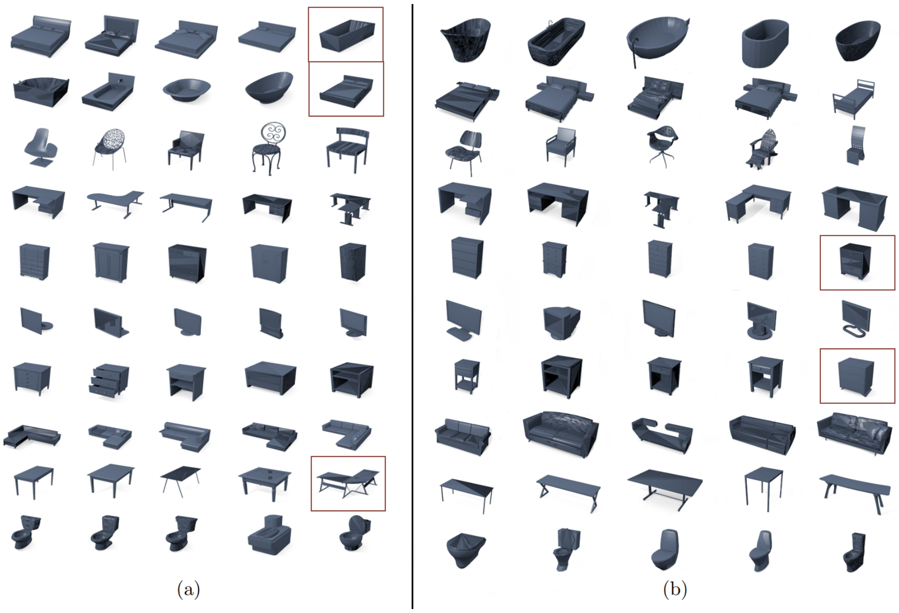


Figure 7. Shape retrieval results by our ACGAN-3D (a) and VACWGAN-GP (b). Random query shape from each class (first column) and the top four matches from the dataset (other columns). Misclassified examples are squared.

As a continuation of this study, application areas such as model repairing and completion using the generative model we propose may be interesting. Extending the classifiers we propose to perform 3D point cloud segmentation would be worthwhile future work. Also, our method in its current form is not able to segment 3D point clouds into semantically meaningful object classes, a limitation we plan to address as future work.

Acknowledgement

This work was supported by TÜBİTAK under the project EEEAG-119E572.

References

[1] Nguyen A, Le B. 3D point cloud segmentation: A survey. 6th IEEE conference on robotics, automation and mechatronics 2013; 225-230.

[2] Hang S, Subhransu M, Kalogerakis E, Learned-Miller E. Multi-view convolutional neural networks for 3d shape recognition. Proc. IEEE International Conference on Computer Vision 2015; 945-953.

[3] Charles Q, Hao S, Mo K, Guibas L. Pointnet: Deep learning on point sets for 3d classification and segmentation. Proc. IEEE Computer Vision and Pattern Recognition 2017; 652-660.

[4] Zhirong W, Song S, Khosla A, Yu F, Zhang L et al. 3d shapenets: A deep representation for volumetric shapes. Proc. IEEE Computer Vision and Pattern Recognition 2015; 1912-1920.

[5] Zhizhong H, Liu Z, Han J, Vong C-M, Bu S et al. Unsupervised 3D local feature learning by circle convolutional restricted Boltzmann machine. IEEE Transactions on Image Processing 2016; 25 (11): 5331-5344.

[6] Zhou Y, Tuzel O. Voxelnet: End-to-end learning for point cloud based 3d object detection. Proc. IEEE conference on computer vision and pattern recognition 2018; 4490-4499.

[7] Caltagirone L, Scheidegger S, Svensson L, Wahde M. Fast LIDAR-based road detection using fully convolutional neural networks. IEEE intelligent vehicles symposium 2017; 1019-1024.

[8] Abbasi A, Kalkan S, Sahillioğlu Y. Deep 3D semantic scene extrapolation. The Visual Computer 2019; 35 (2):271–279.

[9] Kim E, Medioni G. Urban scene understanding from aerial and ground LIDAR data. Machine Vision and Applications 2011; 22 (4):691-703.

[10] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville C. Improved training of wasserstein gans. Advances in neural information processing systems 2017; 30.

[11] Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier gans. International conference on machine learning 2017; 2642-2651.

[12] Özbay E, Çınar A, Güler Z. Structured Deep Learning Supported with Point Cloud for 3D Human Pose Estimation. Proceedings of the 1st International Symposium on Multidisciplinary Studies and Innovative Technologies 2017, 2-4.

[13] Özbay E, Cinar A. A Comparative Study of Object Classification Methods Using 3D Zernike Moment on 3D Point Clouds. Traitement du Signal 2019; 36 (6):549-555.

[14] Qi C, Yi L, Su H, Guibas L. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems 2017; 30.

[15] Song W, Liu Z, Tian Y, Fong S. Pointwise CNN for 3D Object Classification on Point Cloud. Journal of Information Processing Systems 2021; 17 (4):787-800.

[16] Yao X, Guo J, Hu J, Cao Q. Using deep learning in semantic classification for point cloud data. IEEE Access 2019; 7:37121-37130.

[17] Maturana D, Scherer S. Voxnet: A 3d convolutional neural network for real-time object recognition. IEEE/RSJ International Conference on Intelligent Robots and Systems 2015; 922–928.

[18] Shi B, Bai S, Zhou Z, Bai X. Deeppano: Deep panoramic representation for 3-d shape recognition. IEEE Signal Processing Letters 2015; 22 (12):2339-2343.

[19] Kumawat S, Raman S. Lp-3dcnn: Unveiling local phase in 3d convolutional neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019; 4903-4912.

[20] Khan S, Guo Y, Hayat M, Barnes N. Unsupervised primitive discovery for improved 3D generative modeling. Proc. IEEE Computer Vision and Pattern Recognition 2019; 9739-9748.

[21] Misra I, Girdhar R, Joulin A. An end-to-end transformer model for 3d object detection. Proceedings of the IEEE International Conference on Computer Vision 2021; 2906-2917;.

[22] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L et al. Attention is all you need. Advances in neural information processing systems 2017; 30.

[23] Lu D, Xie Q, Gao K, Xu L, Li J. 3DCTN: 3D convolution-transformer network for point cloud classification. IEEE Transactions on Intelligent Transportation Systems 2022.

[24] Yuan W, Held D, Mertz C, Hebert M. Iterative transformer network for 3d point cloud. arXiv preprint arXiv:1811.11209 2018.

[25] Zhang J, Chen L, Ouyang B, Liu B, Zhu J et al. Pointcutmix: Regularization strategy for point cloud classification. Neurocomputing 2022; 505:58-67.

[26] Gschwandtner M, Kwitt R, Uhl A, Pree W. BlenSor: Blender sensor simulation toolbox. International Symposium on Visual Computing 2011; 199-208.

[27] Sahillioğlu Y. Recent advances in shape correspondence. The Visual Computer 2020; 36 (8):1705-1721.

[28] Afham M, Dissanayake I, Dissanayake D, Dharmasiri A, Thilakarathna K et al. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2022; 9902-9912.

[29] Liu Y, Li Z, Zhou C, Jiang Y, Sun J et al. Generative adversarial active learning for unsupervised outlier detection. IEEE Transactions on Knowledge and Data Engineering 2019; 32 (8):1517-1528.

[30] White T. Sampling generative networks. arXiv preprint arXiv:1609.04468 2016.

[31] Sahillioğlu Y, Sezgin M. Sketch-based articulated 3d shape retrieval. IEEE computer graphics and applications 2017; 37 (6):88-101.

[32] Sahillioğlu Y, Kavan L. Detail-preserving mesh unfolding for nonrigid shape retrieval. ACM Transactions on Graphics 2016; 35 (3):1-11.