



Article

3D Point Cloud Shape Generation with Collaborative Learning of Generative Adversarial Network and Auto-Encoder

Dong Yang^{1,2}, Jingyuan Wang¹ and Xi Yang^{1,*}

¹ State Key Laboratory of Integrated Services Networks, School of Telecommunications Engineering, Xidian University, Xi'an 710071, China; yangd@cast504.com (D.Y.); jingyuanw@stu.xidian.edu.cn (J.W.)

² Xi'an Institute of Space Radio Technology, Xi'an 710100, China

* Correspondence: yangx@xidian.edu.cn

Abstract: A point cloud is a simple and concise 3D representation, but point cloud generation is a long-term challenging task in 3D vision. However, most existing methods only focus on their effectiveness of generation and auto-encoding separately. Furthermore, both generative adversarial networks (GANs) and auto-encoders (AEs) are the most popular generative models. But there is a lack of related research that investigates the implicit connections between them in the field of point cloud generation. Thus, we propose a new bidirectional network (BI-Net) trained with collaborative learning, introducing more priors through the alternate parameter optimizations of a GAN and AE combination, which is different from the way of combining them at the network structure and loss function level. Specifically, BI-Net acts as a GAN and AE in different data processing directions, where their network structures can be reused. If optimizing only the GAN without the AE, there is no direct constraint of ground truth on the generator's parameter optimization. This unique approach enables better network optimization and leads to superior generation results. Moreover, we propose a nearest neighbor mutual exclusion (NNME) loss to further homogenize the spatial distribution of generated points during the reverse direction. Extensive experiments were conducted, and the results show that the BI-Net produces competitive and high-quality results on reasonable structure and uniform distributions compared to existing state-of-the-art methods. We believe that our network structure (BI-Net) with collaborative learning could provide a new promising method for future point cloud generation tasks.

Keywords: 3D point cloud; 3D object generation; generative adversarial network; deep learning



Citation: Yang, D.; Wang, J.; Yang, X. 3D Point Cloud Shape Generation with Collaborative Learning of Generative Adversarial Network and Auto-Encoder. *Remote Sens.* **2024**, *16*, 1772. <https://doi.org/10.3390/rs16101772>

Academic Editor: Sara Gonizzi Barsanti

Received: 23 March 2024

Revised: 10 May 2024

Accepted: 14 May 2024

Published: 16 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Three-dimensional point clouds are becoming more and more popular for depicting real-life objects, and thus, they play significant roles in vision, robotics, augmented reality and virtual reality tasks. Among these tasks, learning to generate 3D data has attracted much attention and has been studied using various methods, e.g., image-to-point cloud, image-to-mesh, point cloud-to-voxel, point cloud-to-point cloud, etc. These generated data can be used for different 3D computer vision tasks, such as reconstruction [1–4], completion [5–7], segmentation [8,9], object detection [10–14], classification [15,16] and upsampling [17–21]. However, there are few works tackling the task of generating 3D point clouds from noises, which can create additional training data for recognition, synthesizing new shapes, etc.

To address the point cloud generation task, researchers have proposed several generative adversarial network (GAN)-based models [22–26], auto-regressive models [27], flow-based models [28] and probabilistic generative model [29]. It is obvious that more researchers tend to use GAN-based methods due to their strong generative ability and easy expandability. Meanwhile, we find that many point cloud generation methods explore the effectiveness of their methods in auto-encoding tasks. For instance, Achlioptas et al. [22]

proposed a latent-space GAN (*I*-GAN) to first train an AE to learn a latent representation z and then sample z from its fixed latent space P_z as the input of the GAN. And once the training of the GAN was over, they converted a code learned by the generator into a point cloud using the AE's decoder. Yang et al. [28] also proposed a flow-based auto-encoder to tackle the point cloud generation task, and they also evaluated the auto-encoding performance of their method. The above researchers illustrate the strong correlation between point cloud generation and auto-encoding tasks. Moreover, in the field of image generation, GANs [30–33] and AEs [34–36] are both the most popular generative frameworks. However, all the aforementioned methods use different networks for auto-encoding and generation, and train them individually. This will inevitably ignore the implicit connection between GANs and AEs, which involves valuable priors.

To solve the above problems, this paper proposes a bidirectional network (BI-Net) with a novel training pipeline for tackling the point cloud generation task. The generated multi-category shapes illustrated in Figure 1 demonstrate the robustness of our BI-Net, since various categories of point clouds can be generated with BI-Net in high quality. We deeply explored the relationship between a GAN and AE. In a nutshell, we designed a novel network that could be reused in the reverse training direction, and the network performs generation and auto-encoding tasks in different training directions. Crucially, we trained a GAN and AE alternately in the same training period, allowing the two tasks to share and mutually assist each other in terms of parameters. Coupled with the different object functions of the GAN and AE, the BI-Net was enhanced, thus being capability of generating high-quality shapes.

Similar to a 3D GAN, the BI-Net consists of two modules, i.e., an **En-Di** module and **De-Ge** module. In the forward direction, the BI-Net is considered an AE. The En-Di module is an encoder in the forward direction, and it extracts latent features of the input real point clouds. The features would then be fed to the De-Ge module, which is a decoder in this direction. In the reverse direction, the BI-Net is considered a GAN. The De-Ge module takes noises as input and generates fake point clouds in this direction. The En-Di module plays a discriminator role and discriminates fake/real point clouds in a classification form. Since the De-Ge module learns a mapping function from the latent features to point clouds when training the AE, its non-linear fitting ability was enhanced when training the BI-Net as a GAN, thus achieving the purpose of collaboration between the GAN and AE. Since there is no ground truth in the point cloud generation task, the Chamfer distance (CD) loss and Earth mover's distance (EMD) loss cannot be applied to constrain the quality of the generated point clouds. But in the auto-encoding task, there do exist ground truths, and of course, the CD/EMD loss can be utilized. Therefore, the collaborative learning of the GAN and AE brings more priors, in terms of parameters, into the whole BI-Net.



Figure 1. Examples of multi-category unsupervised shapes generated by BI-Net (e.g., guitar, bag, table, bottle, car, bathtub, bowl, airplane and chair).

To further improve the quality of the generated points, we also propose a nearest neighbor mutual exclusion (NNME) loss for the 3D GAN. Li et al. [20] and Yu et al. [19] proposed the uniform loss and repulsion loss in upsampling tasks, respectively. However, these losses are not applicable to pure generation tasks. The uniform loss utilizes the farthest sampling strategy to choose seed points; because the fake point clouds are not fixed at the beginning, the selected seed points cannot effectively represent shapes. Hence, it is difficult for the network to optimize uniform loss in pure generation tasks. In repulsive loss, there is a parameter relative to the upsampling rate. However, there is no upsampling rate existing in this pure point cloud generation task, so repulsion loss is not applicable, either. To this end, our NNME loss imposes a weak constraint on the point distribution uniformity. It selects non-repetitive minimum distances among points and attempts to minimize the variance of these distances.

Our contributions are three-fold, and are as follows:

- To tackle the point cloud generation task, we propose a bidirectional network with collaborative learning in a novel point cloud generation and auto-encoding pipeline, and the collaborative learning of the GAN and AE brings more priors, in terms of parameters, into the BI-Net.
- We propose a nearest neighbor mutual exclusion loss, which minimizes the variance of the minimum distances to distribute points more uniformly.
- We conducted extensive experiments, including visual and quantitative experiments. The BI-Net shows an excellent generative ability in many categories of objects. The BI-Net produces competitive quantitative generation results compared to existing state-of-the-art methods.

2. Related Work

2.1. Generating 3D Data

Generally, there are four kinds of tasks related to generating 3D data: upsampling, completion, reconstruction and generation. Point cloud upsampling tasks generate dense point clouds from the given sparse point clouds, thus modeling the geometry of the shape more finely. Point cloud completion tasks aim to generate the entire shape from partial input point sets. Reconstruction tasks usually involve the reconstruction of image-to-point cloud, image-to-mesh, image-to-voxel, point cloud-to-mesh data. Generation tasks focus on generating point clouds from noises. It is a newly developing task, and researchers usually call this task point cloud generation or shape generation.

Point cloud upsampling. Yu et al. [19] proposed the PU-Net architecture based on PointNet++ to upsample point clouds; it captures mixing and blending point features in a learned feature space to work on patches in order to expand point sets. Qian et al. [17] first proposed a geometric-based upsampling strategy that could jointly generate coordinates and normals for the generated point sets. PU-GAN [20] was the first work that utilized a GAN to upsample point sets. It uses an up-down-up unit to expand point sets in the generator and utilizes a self-attention mechanism in the discriminator. Qian et al. [18] analyzed the local geometry of point clouds and designed a flexible point cloud upsampling network (FlexiblePU) to adaptively learn interpolation weights and the high-order refinements. The greatest strength of FlexiblePU is that it can upsample point clouds with a flexible factor after one-time training.

Point cloud completion. Rao et al. [37] designed a bidirectional reasoning strategy to learn the patterns in both local-to-global and global-to-local directions. SpareNet [38] presented a channel-attentive EdgeConv to extract the local and global structures of point features, and then projected the completed points to depth maps with a differentiable renderer and used adversarial training to advocate the perceptual realism under different viewpoints. PF-Net [39] utilizes a feature point-based multi-scale generating network to estimate a missing point cloud hierarchically. Fei et al. [40] employed a dual-channel Transformer and cross-attention for point cloud completion, convening the geometric

information from the local regions of incomplete point clouds for the generation of complete ones at different resolutions.

3D shape reconstruction. Fan et al. [41] introduced the PointOutNet structure to address the point cloud reconstruction task from a single image, generating point cloud coordinates. Wu et al. [1] utilized a 3D CNN to build a 3D GAN to construct 3D objects from images by learning the probabilistic latent spaces of two different representations. Gao et al. [2] proposed DEFTET, which optimizes both vertex placement and occupancy when using volumetric tetrahedral meshes for tackling reconstruction problems. Choy et al. [3] proposed a 3D recurrent reconstruction neural network (3D-R2N2) to reconstruct a voxel model from single- or multi-view image(s) and unified single- and multi-view 3D reconstruction in a single framework.

Point cloud generation. It is also called shape generation. Achlioptas et al. [22] first researched this problem. They designed a simple GAN with several MLPs in the generator and discriminator, respectively. Meanwhile, they proposed several evaluation metrics for the 3D GAN quality. Valsesia et al. [23] utilized graph convolution operations to enhance the generator. TreeGAN [24] uses a tree-structured graph convolution network (GCN) to learn the information of father nodes. MRGAN [42] uses a GAN with multi-roots to generate point sets with unsupervised part disentanglement. In addition to GAN-based methods, there are also some methods based on probability theory. Cai et al. proposed ShapeGF [25], which performs stochastic gradient ascent on an un-normalized probability density to move randomly sampled points to high-density areas, near the surface of a certain shape. Luo et al. [29] considered the conversion of points from a noise distribution to a point cloud of a certain shape as the inverse process of the diffusion of particles from a specific distribution to a noise distribution in a thermal system in contact with a heat bath. And they proposed to model this inverse diffusion process as a Markov chain conditioned on a specific shape. Yang et al. [28] proposed PointFlow, which generates point sets by modeling them as a distribution of distributions under a principled probabilistic framework. Kimura et al. [43] proposed a flow-based model ChartPointFlow, which forms a map conditioned on a label. ChartPointFlow maintains the topological structure of the shape clearly. Li et al. [44] presented a modified Variational auto-encoder to achieve parts-aware edition and the generation of point clouds in an unsupervised manner. Several methods have demonstrated their effectiveness on the auto-encoding task through using AE architectures, e.g., *l*-GAN [22], ShapeGF [25] and DPM [29]. Although researchers have made progress in point cloud generation, the implicit connection between a GAN and AE, which contains a substantial prior information gain, is neglected.

2.2. Combination of GAN and AE

The combination of a GAN and AE in the form of network structure or training has been widely studied in the 2D image vision field. Donahue et al. [45] proposed an unsupervised feature learning framework, BiGAN, to learn more useful feature representations for auxiliary problems related to semantics. Besides the generator G and discriminator D of a GAN, BiGAN includes an encoder E to map data generated by G for a latent representation. BiGAN achieves joint training through the D , discriminating jointly in data and latent spaces. To tackle the domain generalization problem of images, Li et al. [46] presented a framework based on adversarial auto-encoders that consisted of an auto-encoder, discriminator and classifier. The discriminator introduces adversarial loss in the auto-encoding process, matching the distribution aligned from different domains by the AE to an arbitrary prior distribution. To alleviate mode collapse and gradient vanishing, Tran et al. [47] proposed Dist-GAN, which combines a GAN and AE in one framework by constraining the generator with the AE. Dist-GAN considers the reconstructed data from the AE as real data for the discriminator, combining the convergence of the AE with that of the discriminator. Thus, Dist-GAN slows down the convergence of the discriminator and alleviates gradient vanishing effectively. An et al. [48] designed AE-OT-GAN, which consists of an auto-encoder and a discriminator,

to alleviate the mode collapse problem in the image generation field. In AE-OT-GAN, the AE and GAN share the same module, which plays the role of the decoder and generator in the AE and GAN, respectively. Although the effectiveness and potential of the combination of a GAN and AE in the field of 2D image vision has been confirmed through extensive research, there is still a lack of related research in the field of 3D point cloud generation.

2.3. Point Distribution Uniformity Loss

During the experiments, we noticed that lots of points in the point cloud generated by TreeGCN are not uniformly distributed, which slightly reduces the generation performance of the BI-Net. Also, the uniformity of the point distribution is an important factor influencing the quality of the point clouds, and many researchers have studied this problem. Yu et al. [19] proposed repulsion loss in PU-Net. This loss is formulated as follows:

$$L_{rep} = \sum_{i=0}^{\hat{N}} \sum_{i' \in K(i)} \eta(\|x_{i'} - x_i\|)w(\|x_{i'} - x_i\|), \quad (1)$$

where the parameter $\hat{N} = rN$ is the number of output points, η and w are formulated as $\eta(r) = -r$ and $w(r) = e^{-\frac{r^2}{h^2}}$, and $K(i)$ is the index set of the k -nearest neighbors of point x_i . Both η and w are relative to the upsampling rate in the upsampling tasks. Hence, the repulsion loss cannot be used in the GAN. Li et al. [20] proposed a uniform loss for PU-GAN. Different from repulsion loss, this loss does not require the upsampling rate. However, it needs to select several seed points through the farthest-point sampling strategy. Hence, this loss heavily relies on the quality of selected points, which is also not suitable for the GAN because the points generated by GANs are very unstable. Similarly, Wen et al. [49] designed a novel loss that uses an averaged local density of points over a large surface area of the point cloud. These losses impose strong constraints on the point distribution uniformity.

3. Proposed Method

3.1. Method Overview

The training pipelines of point cloud generation, auto-encoding and the proposed BI-Net with collaborative learning are illustrated in Figure 2. As shown in Figure 2a, the general pipeline used to tackle the shape generation task involves estimating the point clouds generated by the generator and then discriminating their realness using the discriminator. The adversarial loss is utilized to constrain the generator and discriminator, making them play the game, thereby ensuring the quality of generated point clouds. However, using only the adversarial loss is not enough for high-quality point cloud generation [22]. And for the generation task, an unsupervised learning task, it is impossible to introduce the constraint directly on the distance between the generated point cloud and its ground truth. And as illustrated in Figure 2b, the general pipeline of training the AE involves utilizing the encoder to capture the latent code of input point clouds and reconstructing the input using the decoder. It is feasible to apply constraints on the similarity of the generated point cloud and the ground truth in the process of training the AE.

In particular, the encoder of the AE and the discriminator of the GAN both actually reduce the dimension. Although they end up with different dimensions of low-dimensional data, we can achieve the simultaneous acquisition of both dimensions through the design of the network of the encoder. The decoder of the AE and the generator of the GAN both increase the dimensions, and they finally reach the same dimension of data. As illustrated in Figure 2c, we take advantage of the above traits to integrate the GAN and AE in a single pipeline with collaborative learning. Moreover, to further improve the quality of the generated point clouds, we propose an NNME loss that constrains the uniformity of the point cloud distribution, which alleviates the problem of the local density inconsistency in the generated point clouds. The error backward propagation of the BI-Net is illustrated in

Figure 3. The BI-Net propagates point cloud reconstruction loss and optimizes the En and De in step k and then propagates the GAN loss and NNME loss to the Ge, and the GAN loss to the Di in step $k + 1$, thus achieving the collaborative learning of the BI-Net.

Furthermore, the BI-Net plays different roles in different directions of network computing. In the forward direction, the BI-Net works as an AE, which attempts to learn a mapping function from real points to latent features. Its encoder module takes real points as input and learns a latent feature of the input by encoding it. The latent feature, which is of the same dimensional data as the input noise of the GAN, is then passed through the decoder module to recover the input real points. In the reverse direction, the decoder module of the AE plays the generator module role of the GAN, and the encoder module plays the role of discriminator in the GAN. The network architecture is not changed and the weight learned by the AE is reused. The generator takes N -dim noises as input and attempts to generate fake points. The fake points are then input into the discriminator for the classification task. Meanwhile, when training the generator, the proposed NNME loss is jointly optimized.

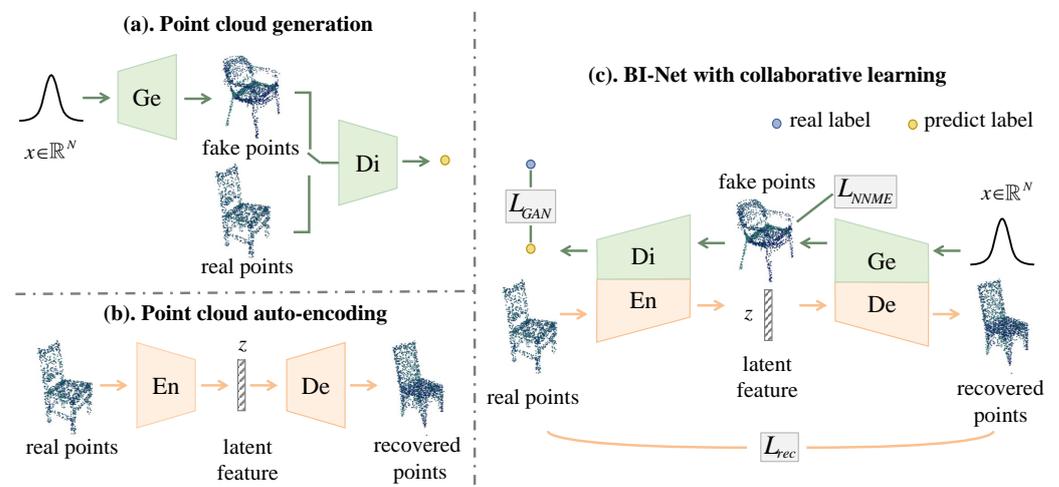


Figure 2. Training pipelines of general point cloud generation, auto-encoding and our BI-Net. N presents the dimension of the input random vector x , and z refers to the latent feature. The graphs and arrows marked in green (orange) present the modules and data flow of the GAN (AE). (a) **Point cloud generation:** Ge and Di represent the generator and discriminator modules of the GAN, respectively. (b) **Point cloud auto-encoding:** En and De represent the encoder and decoder modules of the AE, respectively. (c) **BI-Net with collaborative learning:** generating and recovering point clouds alternatively and optimizing the BI-Net with reconstruction losses L_{rec} and $L_{GAN} + L_{NNME}$, respectively.

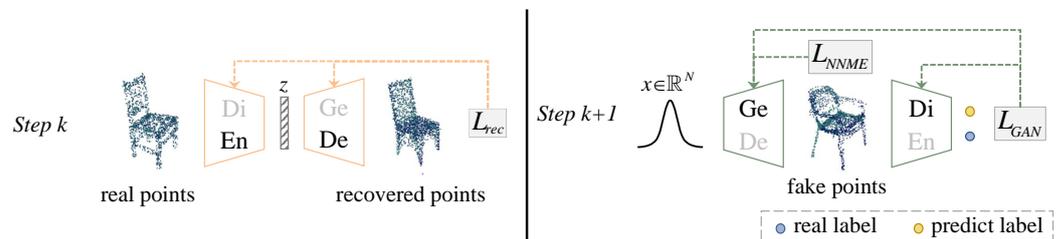


Figure 3. The error backward propagation of the BI-Net. For $k \in [0, N_{epoch}]$, N_{epoch} refers to the number of training epochs. The BI-Net propagates the error and optimizes as an AE (Step k) and GAN (Step $k + 1$) alternatively in the training phase.

3.2. Network Architecture

As mentioned before, the encoder and discriminator perform a similar information compression. The encoder of the AE is used to learn the latent feature of point clouds, of which their dimensions are equal to those of the input noise of the GAN, and the discriminator of the GAN attempts to distinguish whether the input point clouds are real or fake. We slightly modified the En–Di output layer to achieve both goals. The architecture of the En–Di is shown in Figure 4. A point cloud with a size of 2048×3 will first pass through 5 convolutional layers with a 1×1 kernel. And after the max pooling layer, the feature will pass through 3 fully connected layers to obtain a latent feature, which will be input into the last fully connected layer. The last fully connected layer infers the 1 dim discriminating results of the current input point cloud based on the latent feature. To sum up, in the forward direction, the latent feature will be returned as the input of the decoder module. While in the reverse direction, the 1 dim feature is captured to distinguish whether the input point cloud is real or fake.

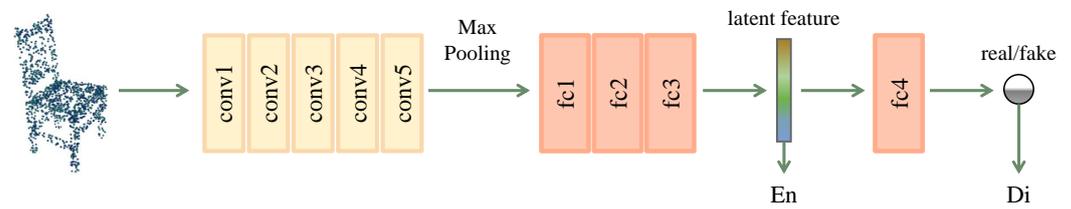


Figure 4. The architecture of the encoder/discriminator. **Conv** and **fc** represent convolutional and fully connected layers, respectively.

The decoder of the auto-encoder takes the N dim latent feature as input and attempts to recover the real points, while the generator of the GAN takes the noises as input and attempts to generate novel unseen points. Both of them perform tasks that increase the dimensions. In other words, the random vector, the input of the generator, can be understood as a kind of random latent feature. Because graph convolutional networks have achieved great success in generating 3D point clouds composed of localized point clusters, we adopted TreeGCN as the backbone of the De–Ge in the BI-Net. The detailed architecture of the decoder/generator is shown in Figure 5. Different from conventional graph convolutional methods, TreeGCN considers points according to the depth of a tree instead of combining features of a static number of points with fixed connectivity. The decoder in the BI-Net builds a tree starting from this latent feature and then grows into a set of nodes through convolution and branching operations. The branching operation creates new children nodes for each leaf on the basis of the current tree, thereby increasing the depth of the tree. The decoder replaces the relationships among neighbors in a graph with the relationships among children nodes and ancestor nodes. Hence, the feature propagation of the decoder is formulated as follows:

$$p_i^{l+1} = \sigma \left(\mathbf{F}_K^l(p_i^l) + \sum_{q_j \in A(p_i^l)} U_j^l q_j + b^l \right), \quad (2)$$

where σ is the activation unit, p_i^l is the i -th node in the graph at the l -th layer, q_j^l is the j -th ancestor node of p_i^l , \mathbf{F}_K^l is a support net (which consists of two MLPs) used to enhance the fitting ability of the MLPs, and $A(p_i^l)$ is the set of all ancestor nodes of p_i^l . U_j^l and b^l are the weight and bias parameters of the l -th layer, respectively. Similarly, in the reverse direction, the generator module takes noises as input and generates fake points, which are input into the discriminator module for classification.

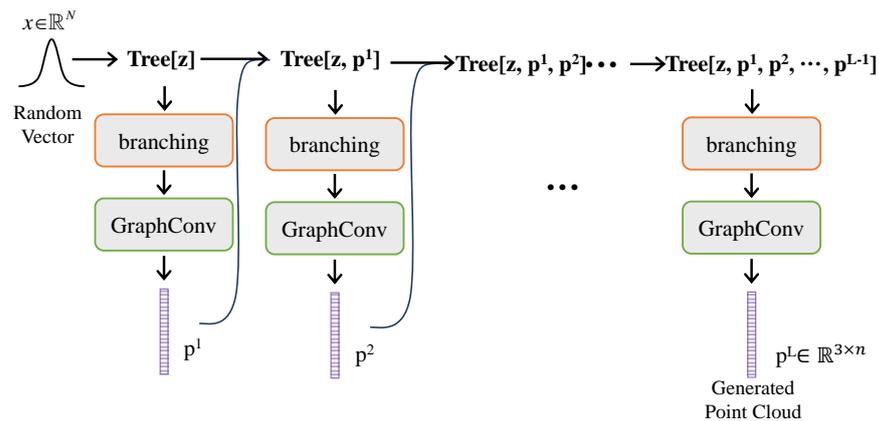


Figure 5. The architecture of the decoder/generator, which is the same as the generator of TreeGAN [24]. **GraphConv** and **branching** represent advanced graph convolution and branching operators, respectively.

3.3. Auto-Encoder Direction

In the forward direction, we train the BI-Net as an AE to reconstruct the input point clouds. Compared with images, point clouds are unordered, and there is no strong relevance of information between different points; therefore, conventional convolution cannot be adopted. Moreover, the data information of point clouds is much less prevalent than that of images; for example, if we consider a 3-channel and 128×128 -pixel image to include 49,152 ($128 \times 128 \times 3$) data, the image has much more data than the commonly used point cloud (2048×3). These two reasons determine that the generator module of the GAN cannot have a complex structure. Most researchers follow this rule, for example, all of the methods in [22–24] use simple MLPs or assemblies of MLPs, which causes the generator to not have sufficient ability to learn complex mapping from noises to non-linear shapes.

Hence, we trained an AE in this direction to introduce more priors of point clouds into the network, constraining the network parameters to a region in which the BI-Net has a stronger nonlinear fitting ability. The constraint on the hypothesis space of the GAN is achieved by optimizing the parameters of the BI-Net during the training process of the AE. The GAN can be optimized to satisfy the discriminative critical points in the hypothesis space of generating high-quality point clouds. To perform the auto-encoding, we adopted the EMD loss to evaluate the similarity between real points $S_{rp} \subseteq \mathbb{R}^3$ and recovered points $S_{recp} \subseteq \mathbb{R}^3$, which is formulated as

$$\begin{aligned} L_{rec} &= d_{EMD}(S_{recp}, S_{rp}) \\ &= \min_{\phi: S_{recp} \rightarrow S_{rp}} \sum_{x_i \in S_{recp}} \|x_i - \phi(x_i)\|_2, \end{aligned} \quad (3)$$

where $\phi: S_{recp} \rightarrow S_{rp}$ represents the learned mapping function. Actually, the Chamfer distance (CD) is another metric method used to evaluate the similarity between two point sets. However, the main goal of the BI-Net is generating rather than auto-encoding; we only utilized the EMD loss to avoid having the BI-Net deviate from the optimal weights for the generation task. The EMD loss captures the point shape structures, encouraging the recovered points to be located closer to the underlying object surfaces.

3.4. GAN Direction

In the reverse direction, we trained the BI-Net as a GAN to generate point clouds from random vectors. Since the generated point clouds are not fixed at the beginning, a too complex discriminator would lead to the unbalance of the GAN. Hence, in the game of the generator and discriminator in point cloud generation, a discriminator that is simpler than the generator is recommended. We adopted a two-output-layer network, that only consists of a few MLPs as the discriminator of the BI-Net.

In order to guarantee the training stability, we selected the object function of the Wasserstein GAN [50]. The loss used to optimize the generator module is defined as follows:

$$L_{gen} = -\mathbb{E}_{z \sim \mathcal{Z}}[D(G(z))], \quad (4)$$

where G and D represent the generator module and discriminator module, respectively. $z \in \mathcal{N}(\mathbf{0}, I)$ represents a noise that is from a normal distribution \mathcal{Z} . The loss of the discriminator module is defined as follows:

$$L_{disc} = \mathbb{E}_{z \sim \mathcal{Z}}[D(G(z))] - \mathbb{E}_{x \sim \mathcal{R}}[D(x)] + \lambda_{gp} \mathbb{E}_{\hat{x}} \left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \right], \quad (5)$$

where \hat{x} is sampled from the line segments between real point sets x and generated fake point sets $G(z)$. \mathcal{R} represents the distribution of real point sets. To prevent collapse during the training process, we utilized a gradient penalty to satisfy the 1-Lipschitz condition, where λ_{gp} is a parameter that controls the weight of the gradient penalty.

3.5. Nearest Neighbor Mutual Exclusion Loss

In the course of the experiment, we found that the point clouds generated by TreeGCN have a non-uniform distribution. However, both the EMD loss and GAN loss have no supervision on the point distribution uniformity; moreover, as aforementioned, existing uniformity losses are inapplicable for training a GAN. Hence, we propose the NNME loss, which has a weak constraint on the point distribution uniformity when training the BI-Net in the reverse direction. Similar to like charges repelling, the NNME loss attempts to make points that are too close mutually exclusive. However, the mutual exclusion should not be too strong; otherwise, the distribution of points will be too dispersed and the BI-Net will face difficulty in generating meaningful shapes. In order to balance the mutual exclusion and the underlying shape, the NNME loss only calculates the distance of each point's nearest neighbor, rather than reject more points. Instead of restricting the point-to-point distance as strictly as the repulsion loss, the NNME loss calculates the variance of these distances to prevent excessive differences in the strength of the mutual exclusion forces at different points. And this makes NNME a much weaker constraint on the point distribution.

Ultimately, the NNME loss makes the points in a point cloud distribute more uniformly by distancing each point from its nearest point and reducing the variance of this distance. When training in the GAN phase, we selected the minimum distances in each generated fake point cloud. The selection process is shown in Figure 6. Supposing that there is a shape with 2048 points, in the first step, we randomly select one point and find its nearest neighbor and then save the distance between them in a Minimum Distance Bank and offline the first point. In the second step, the goal is to find the nearest neighbor of the point that is the nearest neighbor point in the first step and to save the minimum distance in the bank. We repeat this process until there are no points left. At this time, all the minimum distances are stored in the Minimum Distance Bank. The variance of all minimum distances stored in the Minimum Distance Bank is calculated as the NNME loss:

$$L_{NNME} = \frac{\sum (d - \bar{d})^2}{n}, \quad (6)$$

where d is the distance saved in the Minimum Distance Bank, and \bar{d} is the mean value of the distances, while n is the number of distances. Finally, the full objective function that we apply to optimize the generator module combines both the GAN loss and NNME loss as follows:

$$L_G = L_{gen} + \lambda L_{NNME}, \quad (7)$$

where λ is a hyperparameter controlling the weight of the NNME loss.

The training algorithm of the BI-Net in two directions is shown in Algorithm 1. The collaborative learning is implemented through alternate forward propagation and the

optimization of the GAN and AE. The BI-Net will first be optimized with the reconstruction loss L_{rec} in the forward direction. In the reverse direction, the BI-Net will be optimized with the generation loss L_G and the discrimination loss L_{disc} .

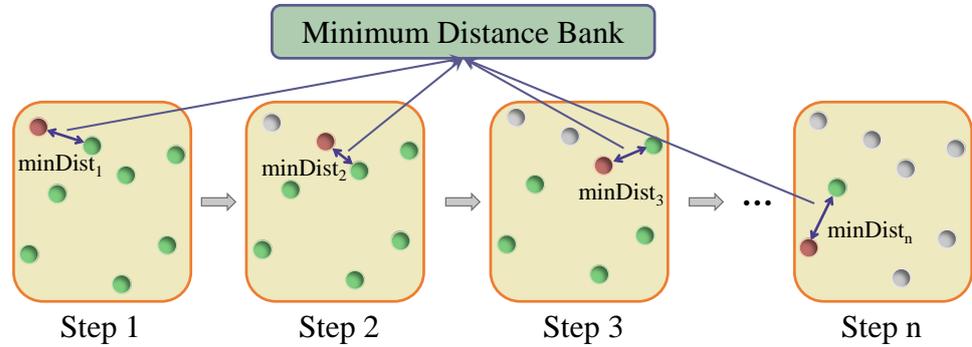


Figure 6. Illustration of selecting the minimum distances in a single point cloud.

Algorithm 1 The training pipeline of the BI-Net

Input: For auto-encoder: batches of real point clouds rp ;

For GAN: batches of noises $z \sim N(0, I)$.

Output: For generator (decoder): batches of fake point clouds fp .

```

1: while not converged do
2:   recover real point clouds,  $recp \leftarrow De(En(rp))$ .
3:    $L_{rec} \leftarrow \min_{\phi: S_{recp} \rightarrow S_{rp}} \sum_{x_i \in S_{recp}} \|x_i - \phi(x_i)\|_2$ 
4:   chain En and De module to form auto-encoder and optimize it with  $L_{rec}$ 
5:   if epoch now > 0: then
6:     With no gradient calculation:
7:     generate fake point clouds,  $fp \leftarrow Ge(z)$ 
8:     discriminate fake and real point clouds,  $flabel \leftarrow Di(fp)$ ,  $rlabel \leftarrow Di(rp)$ 
9:      $L_{disc} \leftarrow \mathbb{E}[flabel] - \mathbb{E}[rlabel] + \lambda_{gp} \mathbb{E}_{\hat{rp}} \left[ \left( \|\nabla_{\hat{rp}} D(\hat{rp})\|_2 - 1 \right)^2 \right]$ 
10:    optimize discriminator module with  $L_{disc}$ 
11:    generate fake point clouds,  $fp \leftarrow Ge(z)$ 
12:     $L_G \leftarrow \lambda \frac{\sum (d-\bar{d})^2}{n} - \mathbb{E}[fp]$ 
13:    optimize generator module with  $L_G$ 
14:   end if
15: end while
16: return  $fp$ 

```

4. Experiments

We implemented the BI-Net with the Pytorch framework. The Adam optimizer was used for both the encoder and decoder modules (i.e., the generator and discriminator modules in the reverse direction.) We take a 128-dimensional random vector as input for the GAN, which means that N was set as 128, as shown in Figure 2. The learning rate was set as 2×10^{-4} , while other coefficients such as the exponential decay rates β_1 and β_2 for the moment estimates in Adam were set as 0 and 0.99, respectively. λ_{gp} and λ_{uni} were set as 10 and 1000, respectively. We adopted LeakyReLU as a non-linearity function without batch normalization. To implement TreeGCN, we used 7 layers with depth degrees [1, 2, 2, 2, 2, 2, 64], and the features of different depths were set as [128, 256, 256, 256, 128, 128, 128, 3]. The loop term of the support network was set as 10.

4.1. Dataset and Evaluation Metrics

We evaluated the BI-Net with ShapeNet [51], which is a large-scale dataset of 3D shapes. We selected chair and airplane shapes for the generation in this evaluation. For the training and testing, we normalized all shapes by centering their bounding boxes to the origin. Moreover, we re-scaled all shapes by a constant, and hence, all coordinates were in

the range $[-1, 1]$, so that the metrics focused on the shapes of the point clouds but not the scale. All of the point clouds in our experiments consisted of 2048 points.

Following prior works, we used the Chamfer distance (CD) and Earth mover's distance (EMD) to quantify the generation quality of the point clouds. For each point in the real point clouds, we first found its nearest neighbor among the generated fake points. The coverage (COV) was measured as the fraction of the generated point clouds that were matched to the real point clouds. Both the CD and EMD can be used to compute the distance. Meanwhile, because coverage cannot indicate exactly how well the covered examples are represented in real point clouds, we used the minimum matching distance (MMD) to measure the fidelity of the real point clouds with respect to the generated points. However, since the low-quality point clouds were most unlikely to match the real point clouds, the MMD was actually less sensitive to the low-quality point clouds. Thus, following PointFlow, we also applied a 1-nearest neighbor accuracy (1-NNA), which was proposed by Lopez-Paz and Oquab [52], to assess whether the distributions of the real and generated point clouds were identical. The 1-NNA can also be computed using both the CD and EMD. Consequently, we utilized four metrics in this study, i.e., COV-CD, COV-EMD, MMD-CD, MMD-EMD, 1-NNA-CD and 1-NNA-EMD.

4.2. Visualization of Auto-Encoder

Figure 7 shows the reconstruction results of the input point cloud during the BI-Net forward auto-encoding process and the generation results of the inverse generating process. As can be observed, although there are still some differences between the reconstructed point cloud obtained using auto-encoding and the input, the underlying shapes are complete and real. The spatial distribution of the reconstructed points is very uniform. To recover the input point cloud, the AE shows much greater interest in the whole point cloud rather than the skeleton of the point cloud, which requires a strong nonlinear fitting ability. Therefore the visualization results illustrate that the auto-encoding task can constrain the network parameters to a region where the BI-Net has a stronger nonlinear fitting ability. The optimization of alternating the GAN and AE allows the BI-Net to generate or reconstruct the underlying geometry of shapes simultaneously.

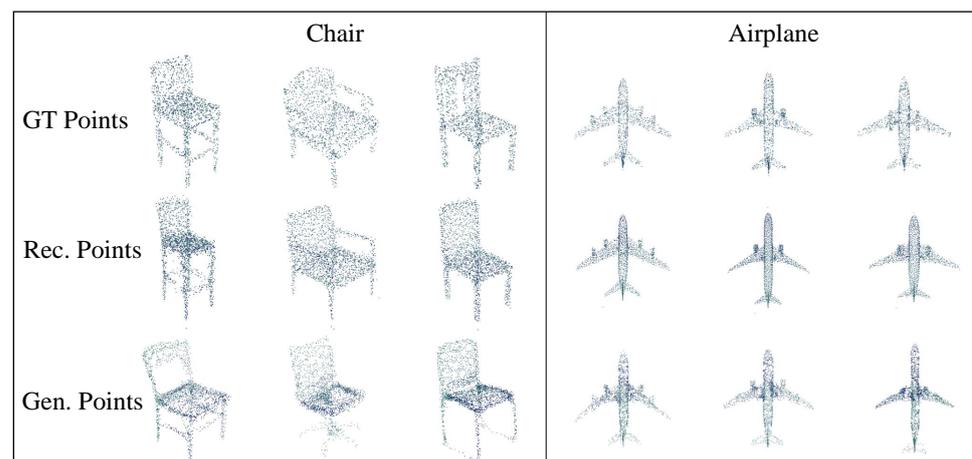


Figure 7. The ground truth, reconstructed shapes and generated shapes of the BI-Net with NNME loss. Rec. Points and Gen. Points represent the reconstructed points and the generated points, respectively.

In addition, we explored the latent space learned by the AE through visualizing the latent features of shapes. We fed the point clouds in the testing set into the encoder of the trained AE to obtain the $b_n \times 1 \times 128$ latent features of point clouds, where b_n refers to the batch size. The latent features produced by the encoder were dimensioned down to 2D, and the clustering of the 2D features was visualized in a 2D plane using t-SNE [53], as presented in Figure 8. It can be observed that for most categories, the latent features

of a certain category are gathered together, and there are significant margins between different categories. The t-SNE clustering performance indicates that the BI-Net is capable of learning effective, category-distinct representations.

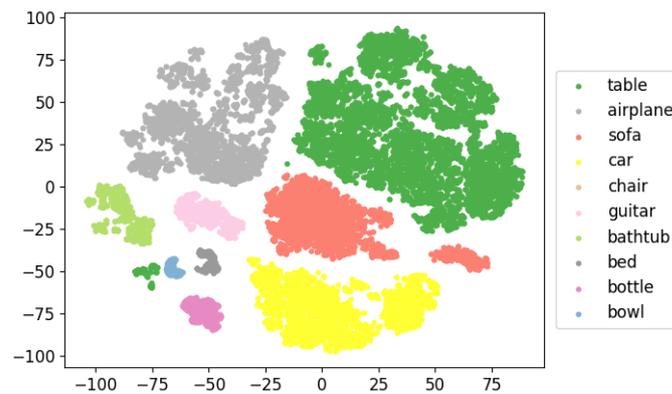


Figure 8. The t-SNE clustering visualization of latent codes obtained from the encoder of the BI-Net.

4.3. Visualization of Generation

We visualize the ground truth and the generation results of TreeGAN, the BI-Net and the BI-Net with the NNME loss for a chair in Figure 9 and airplane in Figure 10. Instead of visualizing the generated point clouds as relatively large points, which is more aesthetically pleasing for visualizing shapes, we set the point size to be small enough to show the uniformity of the point cloud spatial distribution in Figures 9 and 10. It can be observed that the BI-Net can generate the accurate and complete underlying geometry of chair and airplane shapes. For example, in the red boxes of the chairs, the chair legs are oddly shaped or incomplete. In the red boxes of the airplanes, the physical irrationality and asymmetry of the wings is reflected. In addition, it is obvious that there are several agglomerations in the TreeGAN results, as shown in the yellow boxes. While the non-uniform phenomenon has been relieved a lot by the BI-Net, the point distribution is still not totally uniform. In combination with the NNME loss, the point distribution uniformity is acceptable for both the chair and airplane point clouds. The visualization of the generated results strongly supports the validity of the BI-Net with the NNME loss.

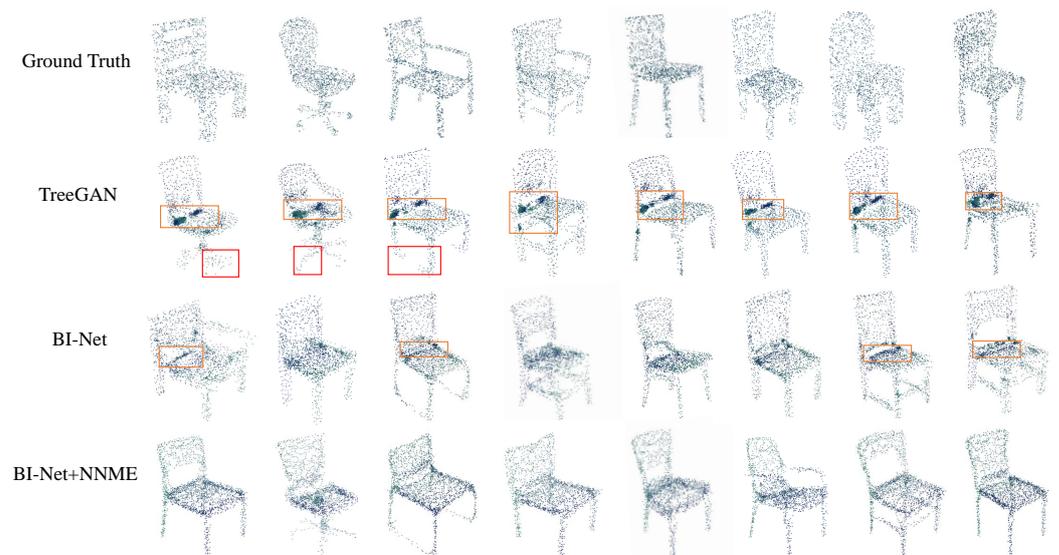


Figure 9. The ground truths of chairs and the corresponding generation shapes of TreeGAN, the BI-Net and the BI-Net with the NNME loss, respectively. The red boxes indicate the shape problems, and the yellow boxes denote the points' local clumping problem.

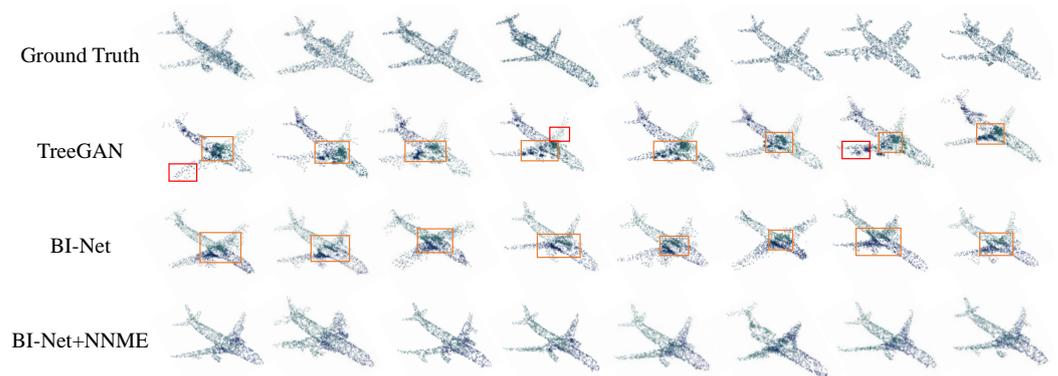


Figure 10. The ground truths of airplanes and the corresponding generation shapes of TreeGAN, the BI-Net and the BI-Net with the NNME loss, respectively. The red boxes indicate the shape problems, and the yellow boxes denote the points local clumping problem.

To validate the generalization ability of the proposed BI-Net, we tested the BI-Net on more categories of shapes in ShapeNet. More examples of the generated point clouds visualization are illustrated in Figure 11, where we present a larger size of points to exhibit the geometry of the generated point clouds more explicitly. The visualization results indicate that the BI-Net can be implemented to generate objects of different underlying shapes, including curved and flat surfaces, e.g., bottle, bag, bathtub, car, guitar, table and bowl shapes. All the visualizations of the generation results demonstrate that the collaborative learning of the GAN and AE led the BI-Net to generate better underlying shapes.

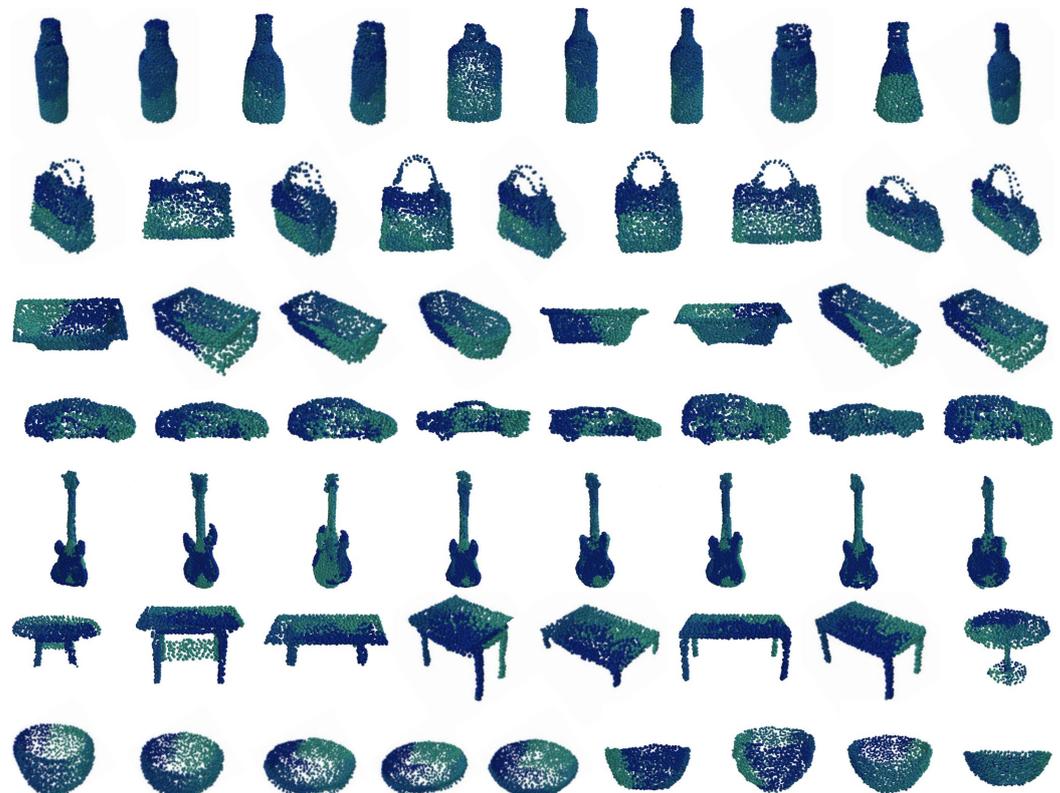


Figure 11. Visualization of shapes generated by the BI-Net for more categories (e.g., bottle, bag, bathtub, car, guitar, table and bowl).

4.4. Ablation Study

We conducted an ablation study to quantify the importance of the proposed BI-Net with collaborative learning (CL) and the NNME loss. Since the generator of the BI-Net is the

same as that of TreeGAN, we used TreeGAN with CL to represent the BI-Net, and TreeGAN with the NNME loss and CL refers to the BI-Net with the NNME loss. The generation results of the chair and airplane shapes are shown in Table 1. AE in Table 1 means simply introducing an auto-encoding loss in the GAN without collaborative learning, like the way in which most works essentially combine a GAN and AE. We can find that in the task of 3D point cloud generation, the collaborative learning method, that is, the BI-Net, is more effective than the previous simple introduction of an AE reconstruction loss. We can see that the NNME loss has a significant effect on improving the shape generation quality of both TreeGAN and the BI-Net. For both chair and airplane generation, the BI-Net without NNME performs better than TreeGAN, which demonstrates that the collaborative learning of the GAN and AE is of great significance for point cloud generation. And with the NNME loss, the generation performance of the BI-Net is further improved. All the metrics show that the BI-Net with the NNME loss achieves the best performance on shape generation. The ablation study demonstrates the effectiveness and superiority of both the NNME loss and the BI-Net with collaborative learning.

Table 1. Ablation study comparing generation performance. \uparrow : the higher the better; \downarrow : the lower the better. *CL* refers to the proposed collaborative learning of the GAN and AE; thus, a modified TreeGAN with *CL* is actually equivalent to our BI-Net. Bold numbers denote the best results.

Shape	TreeGAN	NNME	AE	CL	COV (% , \uparrow)		MMD (\downarrow)	
					CD	EMD	CD (%)	EMD (/10)
Chair	✓				38.02	6.77	1.494	3.613
	✓	✓			45.00	35.00	0.823	1.486
	✓	✓	✓		50.00	50.00	3.565	2.205
	✓			✓	55.00	40.00	0.831	1.869
	✓	✓		✓	60.00	50.00	0.543	1.114
Airplane	✓				39.37	8.40	0.432	1.953
	✓	✓			45.00	45.00	0.295	1.037
	✓	✓	✓		40.00	25.00	1.098	1.543
	✓			✓	56.25	50.00	0.169	1.093
	✓	✓		✓	60.00	65.00	0.294	0.845

4.5. Evaluation of Generation

We quantitatively compared the BI-Net with six state-of-the-art methods in the point cloud generation task, i.e., r-GAN [22], GCN-GAN [23], TreeGAN [24], ShapeGF [25], DPM [29] and ChartPointFlow [43]. Table 2 reports the metrics of the generation results. Without the NNME loss, the COV-CD and MMD-CD metrics show that the BI-Net achieves the best performance with respect to chair shape generation, i.e., 6.06% and 0.397% better than those of DPM. And the MMD-EMD metric of the BI-Net is close to the optimal value of 1.784. For airplane shape generation, the BI-Net obtains the best results for the COV-CD, COV-EMD and MMD-CD metrics. And its MMD-EMD and 1-NNA-CD metrics are also competitive compared with those of SOTAs. These results support the conclusion that the collaborative learning of the GAN and AE helps the network learn how to generate better shapes effectively. However, as is shown in Table 2, the BI-Net shows a poor performance in 1-NNA-EMD, which illustrates that the distribution of shapes generated by the BI-Net is different from that of the real shapes.

With the NNME loss, the generation performance of the BI-Net on most metrics far exceeded those of the existing state-of-the-art methods. For chair shape generation, the BI-Net outperformed the other methods by a large margin with respect to all the COV and MMD metrics. The BI-Net also achieves a comparable performance to those of DPM and ShapeGF under the 1-NNA score. Furthermore, it should be noted that the MMD-CD metric value was reduced to approximately 40% of those of other state-of-the-art methods, which is a considerable improvement. With the NNME loss, the BI-Net also achieves the best performance in all COV metrics, including the MMD-EMD and 1-NNA-CD metrics for

airplane shape generation, and it achieves the second best performance for MMD-CD and 1-NNA-EMD. The generation results demonstrate that both the coverage and fidelity of the points generated by the BI-Net achieve the best performance. Based on the above results, we note that the BI-Net and NNME loss make a lot of sense in generating high-quality point clouds.

Table 2. Shape generation results. \uparrow : the higher the better; \downarrow : the lower the better. Bold numbers denote the best results. The italics are sub-optimal results. The 1-NNA (%) metrics that are closer to 50% are better.

Shape	Model	COV (%), \uparrow		MMD (\downarrow)		1-NNA (%)	
		CD	EMD	CD (%)	EMD (/10)	CD	EMD
Chair	r-GAN	46.23	22.14	1.345	3.104	69.67	100
	GCN	39.84	35.09	1.535	2.213	77.86	95.8
	TreeGAN	38.02	6.77	1.494	3.613	74.92	100
	ShapeGF	48.53	46.71	1.318	1.785	56.17	62.69
	DPM	48.94	47.52	1.228	1.784	60.11	69.06
	ChartPointFlow	-	-	-	-	-	58.31
	BI-Net	55.00	40.00	0.831	1.869	70.00	97.50
	BI-Net+NNME	60.00	50.00	0.543	1.114	65.00	67.50
Airplane	r-GAN	42.17	13.84	0.382	1.81	77.59	98.52
	GCN	39.04	18.62	0.471	1.65	89.13	98.6
	TreeGAN	39.37	8.4	0.432	1.953	83.86	99.67
	ShapeGF	46.79	44.98	0.373	1.061	62.69	71.91
	DPM	48.71	45.47	0.328	1.061	64.83	75.12
	ChartPointFlow	-	-	-	-	-	65.08
	BI-Net	56.25	50.00	0.169	1.093	65.63	93.75
	BI-Net+NNME	60.00	65.00	0.294	0.845	60.00	70.00

5. Conclusions

In this paper, we propose a novel and promising bidirectional network (BI-Net) for tackling the 3D point cloud generation task. In contrast to existing approaches that usually produce point clouds by training a GAN and AE without collaboration, we utilized the network structure similarity and the task similarity between the AE and GAN ingeniously, and we trained a GAN and AE alternately to achieve collaborative learning. The BI-Net was trained toward auto-encoding and generation tasks in the forward and reverse directions, respectively. The alternating training of the network in both directions enhances the non-linear fitting ability of the generator module, and the parameters optimized using the AE help the GAN to step into a better region. In this way, the BI-Net not only enables the integration of prior information significantly, but also enhances the ability to generate reasonably structured and high-quality point clouds. To further improve the uniformity of generated point clouds, we proposed a Nearest Neighbor Mutual Exclusion loss, which imposes a precise constraint during the training process. We conducted extensive experiments to demonstrate that our BI-Net with the collaborative learning and NNME loss obtains the best results for the COV-CD, COV-EMD and MMD-CD metrics, which means that our approach outperforms existing state-of-the-art GAN and AE methods, providing a novel and robust solution to the 3D point cloud generation challenge.

Author Contributions: Methodology, D.Y., J.W. and X.Y.; Software, D.Y. and J.W.; Formal analysis, J.W.; Writing—original draft, D.Y. and J.W.; Writing—review & editing, X.Y.; Funding acquisition, X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62372348; in part by the Shaanxi Outstanding Youth Science Fund Project under Grant 2023-JC-JQ-53; and in part by the Fundamental Research Funds for the Central Universities under Grant QTZX24080.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wu, J.; Zhang, C.; Xue, T.; Freeman, B.; Tenenbaum, J. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Volume 29.
2. Gao, J.; Chen, W.; Xiang, T.; Jacobson, A.; McGuire, M.; Fidler, S. Learning deformable tetrahedral meshes for 3D reconstruction. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020; Volume 33, pp. 9936–9947.
3. Choy, C.B.; Xu, D.; Gwak, J.; Chen, K.; Savarese, S. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 628–644.
4. Li, Y.; Zhao, Z.; Fan, J.; Li, W. ADR-MVSNet: A cascade network for 3D point cloud reconstruction with pixel occlusion. *Pattern Recognit.* **2022**, *125*, 108516. [[CrossRef](#)]
5. Zhang, Z.; Leng, S.; Zhang, L. Cross-domain point cloud completion for multi-class indoor incomplete objects via class-conditional GAN inversion. *ISPRS J. Photogramm. Remote Sens.* **2023**, *206*, 118–131. [[CrossRef](#)]
6. Cheng, M.; Li, G.; Chen, Y.; Chen, J.; Wang, C.; Li, J. Dense Point Cloud Completion Based on Generative Adversarial Network. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–10. [[CrossRef](#)]
7. Hao, R.; Wei, Z.; He, X.; Zhu, K.; Wang, J.; He, J.; Zhang, L. Multistage Adaptive Point-Growth Network for Dense Point Cloud Completion. *Remote Sens.* **2022**, *14*, 5214. [[CrossRef](#)]
8. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
9. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.* **2019**, *38*, 108524. [[CrossRef](#)]
10. Qian, R.; Lai, X.; Li, X. BADet: Boundary-Aware 3D Object Detection from Point Clouds. *Pattern Recognit.* **2022**, *125*, 108524. [[CrossRef](#)]
11. Yang, J.; Shi, S.; Wang, Z.; Li, H.; Qi, X. ST3D: Self-training for unsupervised domain adaptation on 3D object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021.
12. Noh, J.; Lee, S.; Ham, B. HVPR: Hybrid voxel-point representation for single-stage 3D object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021.
13. Feng, M.; Gilani, S.Z.; Wang, Y.; Zhang, L.; Mian, A. Relation graph network for 3D object detection in point clouds. *IEEE Trans. Image Process.* **2020**, *30*, 92–107. [[CrossRef](#)] [[PubMed](#)]
14. Wang, Z.; Zhang, Y.; Luo, L.; Yang, K.; Xie, L. An End-to-End Point-Based Method and a New Dataset for Street-Level Point Cloud Change Detection. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5703015. [[CrossRef](#)]
15. Wu, W.; Qi, Z.; Fuxin, L. Pointconv: Deep convolutional networks on 3D point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9621–9630.
16. Liu, Y.; Cong, Y.; Sun, G.; Zhang, T.; Dong, J.; Liu, H. L3DOC: Lifelong 3D object classification. *IEEE Trans. Image Process.* **2021**, *30*, 7486–7498. [[CrossRef](#)] [[PubMed](#)]
17. Qian, Y.; Hou, J.; Kwong, S.; He, Y. PUGeo-Net: A geometry-centric network for 3D point cloud upsampling. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 752–769.
18. Qian, Y.; Hou, J.; Kwong, S.; He, Y. Deep Magnification-Flexible Upsampling Over 3D Point Clouds. *IEEE Trans. Image Process.* **2021**, *30*, 8354–8367. [[CrossRef](#)] [[PubMed](#)]
19. Yu, L.; Li, X.; Fu, C.W.; Cohen-Or, D.; Heng, P.A. Pu-net: Point cloud upsampling network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2790–2799.
20. Li, R.; Li, X.; Fu, C.W.; Cohen-Or, D.; Heng, P.A. Pu-gan: A point cloud upsampling adversarial network. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7203–7212.
21. Li, T.; Lin, Y.; Cheng, B.; Ai, G.; Yang, J.; Fang, L. PU-CTG: A Point Cloud Upsampling Network Using Transformer Fusion and GRU Correction. *Remote Sens.* **2024**, *16*, 450. [[CrossRef](#)]
22. Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; Guibas, L. Learning representations and generative models for 3D point clouds. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 40–49.
23. Valsesia, D.; Fracastoro, G.; Magli, E. Learning localized generative models for 3D point clouds via graph convolution. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
24. Shu, D.W.; Park, S.W.; Kwon, J. 3D point cloud generative adversarial network based on tree structured graph convolutions. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3859–3868.
25. Cai, R.; Yang, G.; Averbuch-Elor, H.; Hao, Z.; Belongie, S.; Snavely, N.; Hariharan, B. Learning gradient fields for shape generation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 364–381.

26. Sun, Y.; Yan, K.; Li, W. CycleGAN-Based SAR-Optical Image Fusion for Target Recognition. *Remote Sens.* **2023**, *15*, 5569. [[CrossRef](#)]
27. Sun, Y.; Wang, Y.; Liu, Z.; Siegel, J.; Sarma, S. Pointgrow: Autoregressively learned point cloud generation with self-attention. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 61–70.
28. Yang, G.; Huang, X.; Hao, Z.; Liu, M.Y.; Belongie, S.; Hariharan, B. Pointflow: 3D point cloud generation with continuous normalizing flows. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4541–4550.
29. Luo, S.; Hu, W. Diffusion probabilistic models for 3D point cloud generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021.
30. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [[CrossRef](#)]
31. Yi, Z.; Chen, Z.; Cai, H.; Mao, W.; Gong, M.; Zhang, H. BSD-GAN: Branched Generative Adversarial Network for Scale-Disentangled Representation Learning and Image Synthesis. *IEEE Trans. Image Process.* **2020**, *29*, 9073–9083. [[CrossRef](#)] [[PubMed](#)]
32. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
33. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4401–4410.
34. Xu, W.; Keshmiri, S.; Wang, G. Adversarially Approximated Autoencoder for Image Generation and Manipulation. *IEEE Trans. Multimed.* **2019**, *21*, 2387–2396. [[CrossRef](#)]
35. Pidhorskyi, S.; Adjero, D.A.; Doretto, G. Adversarial latent autoencoders. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 14104–14113.
36. Chaurasiya, R.K.; Arvind, S.; Garg, S. Adversarial Auto-encoders for Image Generation from standard EEG features. In Proceedings of the 2020 First International Conference on Power, Control and Computing Technologies (ICPC2T), Raipur, India, 3–5 January 2020; pp. 199–203.
37. Rao, Y.; Lu, J.; Zhou, J. Global-local bidirectional reasoning for unsupervised representation learning of 3D point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 5376–5385.
38. Xie, C.; Wang, C.; Zhang, B.; Yang, H.; Chen, D.; Wen, F. Style-based point generator with adversarial rendering for point cloud completion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021.
39. Huang, Z.; Yu, Y.; Xu, J.; Ni, F.; Le, X. Pf-net: Point fractal network for 3d point cloud completion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 7662–7670.
40. Fei, B.; Yang, W.; Ma, L.; Chen, W.M. DcTr: Noise-robust point cloud completion by dual-channel transformer with cross-attention. *Pattern Recognit.* **2023**, *133*, 109051. [[CrossRef](#)]
41. Fan, H.; Su, H.; Guibas, L.J. A point set generation network for 3d object reconstruction from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 605–613.
42. Gal, R.; Bermano, A.; Zhang, H.; Cohen-Or, D. MRGAN: Multi-rooted 3D shape generation with unsupervised part disentanglement. *arXiv* **2020**, arXiv:2007.12944.
43. Kimura, T.; Matsubara, T.; Uehara, K. Topology-Aware Flow-Based Point Cloud Generation. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 7967–7982. [[CrossRef](#)]
44. Li, S.; Liu, M.; Walder, C. EditVAE: Unsupervised Parts-Aware Controllable 3D Point Cloud Shape Generation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; Volume 36, pp. 1386–1394.
45. Donahue, J.; Krähenbühl, P.; Darrell, T. Adversarial Feature Learning. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
46. Li, H.; Pan, S.J.; Wang, S.; Kot, A.C. Domain generalization with adversarial feature learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5400–5409.
47. Tran, N.T.; Bui, T.A.; Cheung, N.M. Dist-gan: An improved gan using distance constraints. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 370–385.
48. An, D.; Guo, Y.; Zhang, M.; Qi, X.; Lei, N.; Gu, X. AE-OT-GAN: Training GANs from data specific latent distribution. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 548–564.
49. Wen, Y.; Lin, J.; Chen, K.; Chen, C.P.; Jia, K. Geometry-aware generation of adversarial point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 2984–2999. [[CrossRef](#)] [[PubMed](#)]
50. Adler, J.; Lunz, S. Banach wasserstein GAN. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 31.
51. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv* **2015**, arXiv:1512.03012.

-
52. Lopez-Paz, D.; Oquab, M. Revisiting Classifier Two-Sample Tests. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
 53. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.