



## Article

# I-PAttnGAN: An Image-Assisted Point Cloud Generation Method Based on Attention Generative Adversarial Network

Wenwen Li <sup>1,†</sup>, Yaxing Chen <sup>1,\*,†</sup> , Qianyue Fan <sup>1</sup>, Meng Yang <sup>2</sup> , Bin Guo <sup>1</sup> and Zhiwen Yu <sup>1</sup>

<sup>1</sup> Department of Computer Science, Northwestern Polytechnical University, No. 127 West Youyi Road, Xi'an 710072, China; wenwenli@mail.nwpu.edu.cn (W.L.); fanqy@mail.nwpu.edu.cn (Q.F.); guob@nwpu.edu.cn (B.G.); zhiwenyu@nwpu.edu.cn (Z.Y.)

<sup>2</sup> Department of Artificial Intelligence, Xi'an Jiaotong University, No. 28 Xianning West Road, Xi'an 710049, China; mengyang@mail.xjtu.edu.cn

\* Correspondence: yxchen@nwpu.edu.cn

† These authors contributed equally to this work.

**Abstract:** The key to building a 3D point cloud map is to ensure the consistency and accuracy of point cloud data. However, the hardware limitations of LiDAR lead to a sparse and uneven distribution of point cloud data in the edge region, which brings many challenges to 3D map construction, such as low registration accuracy and high construction errors in the sparse regions. To solve these problems, this paper proposes the I-PAttnGAN network to generate point clouds with image-assisted approaches, which aims to improve the density and uniformity of sparse regions and enhance the representation ability of point cloud data in sparse edge regions for distant objects. I-PAttnGAN uses the normalized flow model to extract the point cloud attention weights dynamically and then integrates the point cloud weights into image features to learn the transformation relationship between the weighted image features and the point cloud distribution, so as to realize the adaptive generation of the point cloud density and resolution. Extensive experiments are conducted on ShapeNet and nuScenes datasets. The results show that I-PAttnGAN significantly improves the performance of generating high-quality, dense point clouds in low-density regions compared with existing methods: the Chamfer distance value is reduced by about 2 times, the Earth Mover's distance value is increased by 1.3 times, and the F1 value is increased by about 1.5 times. In addition, the effectiveness of the newly added modules is verified by ablation experiments, and the experimental results show that these modules play a key role in the generation process. Overall, the proposed model shows significant advantages in terms of accuracy and efficiency, especially in generating complete spatial point clouds.

**Keywords:** point cloud generation; multimodal; 3D point cloud map



Academic Editors: Mingyang Zhang, Puzhao Zhang, Xiangming Jiang and Fenlong Jiang

Received: 14 December 2024

Revised: 30 December 2024

Accepted: 31 December 2024

Published: 4 January 2025

**Citation:** Li, W.; Chen, Y.; Fan, Q.; Yang, M.; Guo, B.; Yu, Z. I-PAttnGAN: An Image-Assisted Point Cloud Generation Method Based on Attention Generative Adversarial Network. *Remote Sens.* **2025**, *17*, 153. <https://doi.org/10.3390/rs17010153>

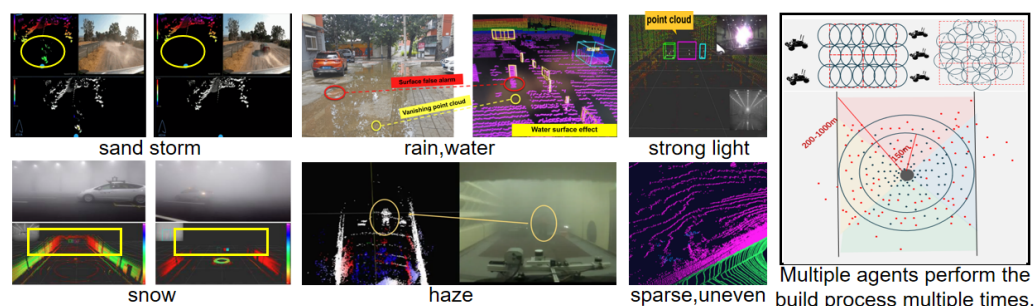
**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Three-dimensional point clouds have important application value in the fields of embodied intelligence, autonomous driving, and virtual reality. With the rapid development of intelligent technology, 3D point cloud map construction has gradually become a research hotspot. However, due to the limitation of sensor hardware, the current 3D point cloud construction technology still has multiple challenges, such as sparse and uneven point cloud data and poor ability to represent long-distance objects.

As shown in Figure 1, in a ground–air collaboration scenario (rightmost), the ground and air agents cooperate to construct large-scale 3D maps. However, the acquired point

cloud data usually have a low overlap area due to the difference in viewpoint between the ground and aerial equipment. The sparsity and inhomogeneity of point cloud data and the insufficient ability to represent distant objects bring significant reconstruction efforts. In addition, when constructing 3D point cloud maps in extreme environments, special substances such as snow, dust, and haze further aggravate the sparsity and inhomogeneity of point cloud data due to their low reflection characteristics from LiDAR signals. These factors weaken the ability of point clouds to represent distant objects and significantly enhance the complexity of map construction, especially in the case of a water surface where there is almost no laser signal reflected; the collected point cloud data are highly sparse, which may have a disastrous impact on the mapping task.



**Figure 1.** Challenges in constructing 3D point cloud maps [1–3].

The construction of 3D point cloud maps often encounters challenges such as sparsity, uneven distribution, and limited perceptual capabilities when dealing with long-distance scenes of point cloud data. These factors can significantly hinder the effectiveness and accuracy of the mapping process. A straightforward solution is to use the agents to repeatedly collect data to fill the target area, which undoubtedly increases the overall task load of the 3D map construction. To this end, a promising alternative is to use point cloud generation methods based on existing point cloud data to infer and generate point cloud regions with sparse and uneven data and poor representation ability. This method can significantly improve the quality and integrity of point cloud data without increasing the burden of additional collection to effectively alleviate the problems of point cloud data sparsity and poor perception ability. Existing point cloud generation methods usually opt for architectures such as generative models (generative adversarial networks (GANs) [4–7] and variational autoencoders (VAE)) [8–10] to reconstruct 3D point clouds from random noise or use Gaussian models based on existing features. These methods have shown good performance in the generation task. However, most methods are limited to generating point cloud data with fixed shapes, simple scenes, or specific resolutions, which have difficulties meeting the complex and diverse requirements of real-world scenes.

Other studies attempt to use image data to generate 3D point clouds and provide necessary prior conditions for point cloud generation by combining multimodal data. Although the quality of the generation has improved, many challenges still remain. Firstly, the generated data are sparse and uneven, ignoring the differences in point cloud data distribution and focusing only on the overall shape, which is challenging to adapt to special scene requirements. Secondly, there is the problem of data redundancy, and the repeated generation of high-resolution and feature-dense regions when resources are limited may lead to waste and increase the task burden. Finally, the point clouds generated by existing generative models on real scenes often have the problems of shape distortion and excessive noise, which limits the practical application value of the generated maps.

In this work, we propose a novel approach to point cloud generation and enhancement, driven by a unique attention mechanism based on point cloud density. Our main contributions can be summarized as follows:

- **Image-aided framework for point cloud generation:** This paper proposes an innovative framework that organically combines image and point cloud data to give full play to the complementary advantages of the two data types. By fusing Generative Adversarial Network (GAN), Variational Autoencoder (VAE) and Normalized Flows (NFs), the prior information provided by imagedata was effectively used, and the texture details and geometric accuracy of the generated point cloud were significantly improved, making the generated results closer to the real scene.
- **Attention generation mechanism for sparse regions:** The model proposes an attention mechanism based on point cloud density, which uses the image input information to generate point clouds for sparse regions, solves the problems of uneven distribution and sparse boundary regions in point cloud data, and effectively reduces the redundancy in high-density areas. Under the premise of maintaining the overall resolution, the proposed mechanism significantly improves the detail representation ability of sparse regions and enhances the global structural consistency and local detail performance.
- **Comprehensive experimental validation:** Experimental results on synthetic datasets (ShapeNet) and real-world datasets (nuScenes) fully demonstrate the effectiveness of the proposed method. Experiments show that the proposed model outperforms existing methods in key metrics such as Chamfer distance (CD), bulldozer distance (EMD), and F1 score, and especially exhibits superior point cloud generation ability in complex real scenes.

The remaining structure of this paper is as follows: Section 2 provides an overview of the related literature; Section 3 explains our proposed methodology in detail; Section 4 presents the results obtained from experiments conducted on the ShapeNet and nuScenes datasets; and finally, in Section 5, we summarize and discuss possible future directions.

## 2. Related Work

The existing point cloud generation works can be roughly divided into four categories: point cloud generation methods based on deep learning, point cloud generation assisted by multimodal data, and point cloud generation based on physical or geometric rules. Next, we elaborate on these methods.

### 2.1. Point Cloud Generation Method Based on Deep Learning

Point cloud generation methods based on deep learning have made significant progress in computer vision and 3D reconstruction in recent years. These methods utilize deep learning models, especially generative adversarial networks (GANs) and variational autoencoders (VAEs), to generate or supplement sparse point cloud data. The advantage of deep learning lies in its ability to automatically learn and extract features from large-scale data, thereby generating higher-quality and more diverse 3D point clouds. In the following are several standard point cloud generation methods based on deep learning.

#### 2.1.1. Generative Adversarial Networks (GANs)

A generative adversarial network (GAN) is a type of deep learning model that generates new data through adversarial training between a generator and a discriminator. In the point cloud generation, the generator aims to create realistic point clouds while the discriminator determines the authenticity of the data. GANs can generate high-quality and diverse point cloud data in complex or sparse scenarios, performing exceptionally well, especially in generating point clouds with intense realism. For instance, PointGAN [4] generates the 3D coordinates of point clouds and combines the local features of each point in the point cloud for training. The generator and the discriminator compete against each

other to produce realistic 3D point clouds. The main objective is for the discriminator to calculate the distribution of the original samples, while the generator's job is to generate new samples from the real data samples. 3D-GAN [7] is a variant of GAN based on voxel data. It achieves multi-view consistent semantic image editing by inferring camera view-points and latent codes. The key of this method is to utilize a pre-trained estimator for better initialization and to use the pixel-level depth calculated from NeRF parameters for better reconstruction of the given image.

### 2.1.2. Variational Autoencoder (VAE)

A variational autoencoder (VAE) is a type of generative model that can learn the latent distribution of data and generate new samples. In the point cloud generation, VAEs map the input point cloud to the latent space through an encoder and then generate new point cloud data from the latent space through a decoder. VAEs can effectively capture the structural information of point cloud data and generate smooth and structurally consistent point clouds. Pointivae [8] is a point cloud generation framework based on a variational autoencoder (VAE) aiming to construct local relationships and enhance generation capabilities. This framework consists of three main components: encoder, flow model, and decoder. The encoder is responsible for aggregating the neighborhood relationships of point clouds and generating high-quality latent space representations, thereby capturing the local structural information of point clouds. The flow model adopts a reversible residual coupling stack (Reversible Residual Coupling Stack), learning complex shape features from latent codes through reversible operations, further improving the quality of generated point clouds. The decoder then reversely decodes the shape's latent codes generated by the flow model, converting them into realistic 3D point cloud data. PointIVAE can effectively model point clouds' local and global structures through this framework, significantly enhancing the ability to generate point clouds. VG-VAE [11] (Venatus Geometric Variational Autoencoder) is a model designed to capture hierarchical local and global geometric features in point clouds without supervision. Recent studies have shown that the underlying geometric structure in point cloud processing is crucial. Our contribution lies in the Geometric Proximity Correlator (GPC) proposal and latent variational sampling techniques to effectively extract and analyze the geometric morphology of point clouds. GPC optimizes the extraction of local geometric features, while global geometric information is captured through variational sampling. Additionally, we combine a simplified hybrid approach of vector algebra and 3D geometry to extract the basic geometric features of each point, thereby providing support for unsupervised learning hypotheses. PCN [12] (Point Completion Network) processes raw point clouds directly without relying on any predefined structural assumptions (like symmetry) or annotations (such as semantic labels) related to the underlying shape. Utilizing a decoder-based architecture, it achieves detailed completions with minimal parameters, ensuring efficient and fine-grained output.

### 2.1.3. Autoregressive Models

An autoregressive model is a generative model where each generated element (such as each point in a point cloud) is conditionally generated, meaning the generation of each point depends on the previously generated points. This model generates data sequentially, typically by recursively predicting the distribution of the next point. Assuming each point in the point cloud is a random variable and is generated in a particular order, the model predicts the distribution of the next point based on the already generated points. This approach enables the gradual refinement of the generated content, ensuring that each point is consistent with the previous ones. Some variants of PixelCNN [13,14] are based on any vector, including descriptive labels or tags or latent embeddings created by other networks.

Moreover, they can generate diverse and realistic scenes representing different animals, objects, landscapes, and structures. When conditioned on embeddings generated by a convolutional network, given a single image of an unseen face, it can generate various new portraits of the same person with different facial expressions, poses, and lighting conditions.

#### 2.1.4. Normalizing Flows

Normalizing Flows map simple distributions (such as Gaussian distributions) to complex distributions through a series of invertible transformations. Flow models can learn any complex probability distribution with these transformations, efficiently accomplishing generative tasks. Specifically, flow models start from a simple initial distribution (such as the standard normal distribution) and, through a series of invertible transformations, gradually approximate the target data distribution. Each transformation step is not only differentiable but also maintains invertibility, which enables the generated data to be mapped back to the latent space through a reverse transformation, thus achieving efficient sampling and training processes. Unlike traditional generative models such as variational autoencoders (VAEs) and generative adversarial networks (GANs), flow models offer an exact probability density estimation, thus demonstrating higher accuracy in both the generation process and sample inference. Flow models can generate more diverse, high-quality point cloud data by stacking multiple transformation layers. In point cloud generation tasks, flow models can generate dense and diverse point clouds by learning the latent distribution of point clouds. Especially when dealing with complex geometric structures, flow models can effectively simulate the complexity of spatial distribution and generate point cloud data that conform to the shapes of real-world objects. RealNVP [15] extends the capabilities of such models by introducing real-valued non-volume preserving (real NVP) transformations, which are a set of powerful, invertible, and learnable transformation methods, thereby forming an unsupervised learning algorithm. This algorithm enables precise log-likelihood calculation and accurate sample generation, supports exact inference of latent variables, and provides an interpretable latent space. In addition, there are numerous variants based on Glow [16–19]. The latent variable model constructed based on the affine coupling layer of the normalizing flow can generate 3D point clouds of any size to represent the given latent shape precisely. This model demonstrates extensive application value in multiple fields, such as generation tasks, autoencoding, and single-view shape reconstruction.

#### 2.2. Multimodal-Assisted Point Cloud Generation

Multimodal-assisted point cloud generation, which fuses data such as images, depth maps, and LiDAR from various sensors, supplements the missing information in a single modality, generating complete, accurate, and diverse point cloud data. This approach is efficient in complex environments and for generating sparse point clouds. Xie et al. [20] proposed a context-aware fusion module that adaptively selected high-quality reconstructed parts (such as table legs) from different coarse 3D volumes to generate the fused 3D volume. Finally, the fused volume was optimized using a refinement condition to produce the final output. Image2Point [21] transfers an image pre-training model to a point cloud model by replicating or dilating weights. With a minimal effort to fine-tune the transformed image-p model (FIP) only on the input, output, and normalization layers, it achieves competitive performance in 3D point cloud classification. In addition, there are some unique methods. For instance, Ziyi Wang et al. [22] achieved the transfer of pre-trained knowledge from the 2D domain to 3D point cloud tasks through Point-to-Pixel Prompting and Pixel-to-Point Distillation. Firstly, Point-to-Pixel Prompting converted point clouds into color images that retained geometric information and processed them using pre-trained 2D image models. Then, Pixel-to-Point Distillation transferred the knowledge of 2D image models to point

cloud models in the feature space, enhancing their inference efficiency and capability. This cross-modal transfer enables point cloud tasks to benefit directly from the pre-trained knowledge of 2D models.

### *2.3. Point Cloud Generation Based on Physical or Geometric Rules*

Point cloud generation methods based on physical and geometric rules generate realistic and accurate point cloud data by simulating the physical processes or geometric structure features of the real world. Among them, the generation methods based on geometric features usually rely on the geometric properties of point cloud data, such as the relative positions between points, standard directions, curvatures, and other spatial geometric features. These methods typically do not depend on images or external information but focus on using existing point cloud data or structural geometric knowledge to generate new point clouds. For instance, GCN assumes that particular objects in space have known geometric shapes (such as circles, squares, etc.), and then point clouds conforming to these shapes can be generated based on these geometric features. By modeling the geometric features of objects, the generated point clouds will naturally follow these constraints in space, ensuring they conform to the physical forms in the real world. Additionally, considering objects' planar, curved, or volumetric characteristics, geometric transformations (such as rotation, translation, scaling, etc.) can be used to generate new point cloud data, satisfying the spatial distribution of these geometric shapes. The normal and curvature of the point cloud are typically exploited to generate more surface details. During the point cloud generation process, by adjusting the normal and curvature features of the point cloud, more similar point clouds can be generated, or more complex structures can be created. Besides the methods for generating point clouds based on geometric rules, there are also methods for generating point cloud data based on physical models. These methods usually rely on the principles of physics (such as mechanics, fluid mechanics, electromagnetism, etc.) to model and predict the structure and form of objects or scenes and then generate point cloud data by simulating objects' physical behavior or physical processes. For example, Vinicius Mikuni et al. [23] proposed a fast point cloud diffusion technique based on diffusion models to address issues such as continuous coordinates, permutation invariance, and random dimensions in particle physics data generation. It achieved high precision while enhancing the generation speed by gradual denoising through diffusion models and introducing progressive distillation techniques to accelerate the generation process. This method could accurately reproduce the complex characteristics of proton–proton collision hadron jets, surpassing existing models and demonstrating the advantages of diffusion models in particle physics data generation. Liu Yang et al. [24] proposed a method called Physical Information Generative Adversarial Network (PI-GAN), which integrated physical laws into the generative adversarial network (GAN) architecture and utilized automatic differentiation and Stochastic Differential Equations (SDEs) to solve forward, inverse, and mixed stochastic problems. PI-GAN employed a Wasserstein GAN (WGAN-GP) [25] to enhance generation stability and combined multiple feedforward deep neural networks (DNNs) and SDE-induced neural networks to generate stochastic processes that conformed to physical laws. Experiments showed that PI-GAN could efficiently generate high-dimensional stochastic processes, especially when dealing with SDE problems up to 30 dimensions, demonstrating outstanding accuracy and effectiveness, with computational costs increasing at a low polynomial rate as sensor data grew.

## **3. Methods**

This study addresses the issues of sparsity and uneven distribution in point cloud data by leveraging prior knowledge from image data. Unlike most point cloud generation

methods, which focus solely on global representation, our approach enhances the representation of distant objects. It improves the accuracy of sparse region representation while simultaneously strengthening the shape features of the point cloud during the generation process. By incorporating a point cloud density-based attention mechanism, the model selectively emphasizes distant objects and sparse regions, thereby capturing global shapes and fine details more accurately.

In 3D map construction, the precision and detail of the map are directly impacted by the quality of the point cloud data. Our method fills in sparse areas and restores missing information, ensuring that even low-overlap or sparse regions contribute to the map with high accuracy, improving the overall map quality and detail restoration process. Experimental results based on publicly available 3D point cloud datasets show that our approach effectively addresses the challenges of sparse data representation and edge detail recovery in real-world scenarios, providing substantial performance improvements. The specific implementation of the method and related experiments are described in the following subsections.

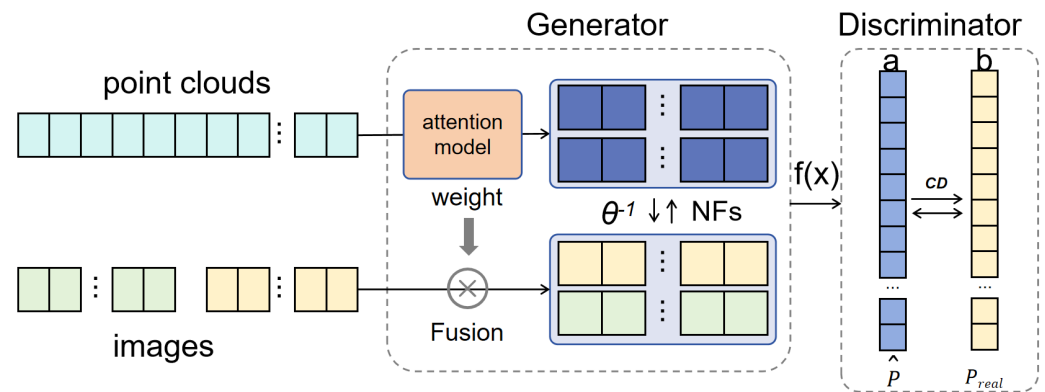
### 3.1. Overall Architecture

The overall architecture of the proposed model integrates multiple components designed to address the challenges in point cloud generation, particularly for sparse and non-uniform regions. The architecture combines elements of generative adversarial networks (GANs), variational autoencoders (VAEs), point cloud density-based attention mechanisms, and normalizing flow models. This multi-module approach allows for the effective generation of high-quality point clouds while accounting for the spatial and structural characteristics of the data. The proposed model consists of two main modules: the generator module and the discriminator module. These modules work in an GAN, where the generator aims to produce realistic point clouds, and the discriminator evaluates their authenticity. Below is an overview of this architecture's key components and functionalities. The generator generates realistic point cloud samples from latent space representations. It incorporates several advanced techniques to improve point cloud generation, particularly for challenging regions with low-density or non-uniform structures. The discriminator's role is to differentiate between real and generated point clouds, providing adversarial feedback to the generator. The generator and discriminator are trained antagonistically, where the generator tries to minimize the gap between generated and real point clouds, while the discriminator **aims to correctly distinguish between them**. This adversarial training process improves the quality of the generated point clouds over time. The model operates as follows:

- The generator produces a point cloud by first encoding latent variables through the VAE and then refining the result using the density-based attention mechanism and normalizing flows.
- The discriminator evaluates the authenticity of the generated point cloud by comparing it to real samples and computing the corresponding loss.
- Both the generator and discriminator are updated iteratively. The generator aims to deceive the discriminator by improving the realism of the generated point clouds, while the discriminator strives to distinguish between real and fake samples correctly.

The proposed architecture provides a robust framework for generating high-quality point clouds by integrating advanced techniques such as GANs, VAEs, attention mechanisms, and normalizing flows. The key advantage of this architecture is its ability to generate point clouds that accurately reflect the structural characteristics of real-world data, especially in regions that are challenging for traditional methods. The overall architecture

is illustrated in Figure 2, which shows the interactions between the generator, discriminator, and the key components involved in the training process.



**Figure 2.** Overall architecture. **In the generation module, the attention weight is calculated according to the point cloud density and weighted fusion is performed with the image data. Then, the Normalized Flows (NFs) model was used to learn the reversible transformation from complex distribution to simple Gaussian distribution. Finally, a Variational AutoEncoder (VAE) was used to model the two distributions as conditional distributions. Method In the discrimination module, the inference results of the generation module are classified and compared, and the gap between the generated point cloud data and the real point cloud data is calculated.**

### 3.2. Point Cloud Generator

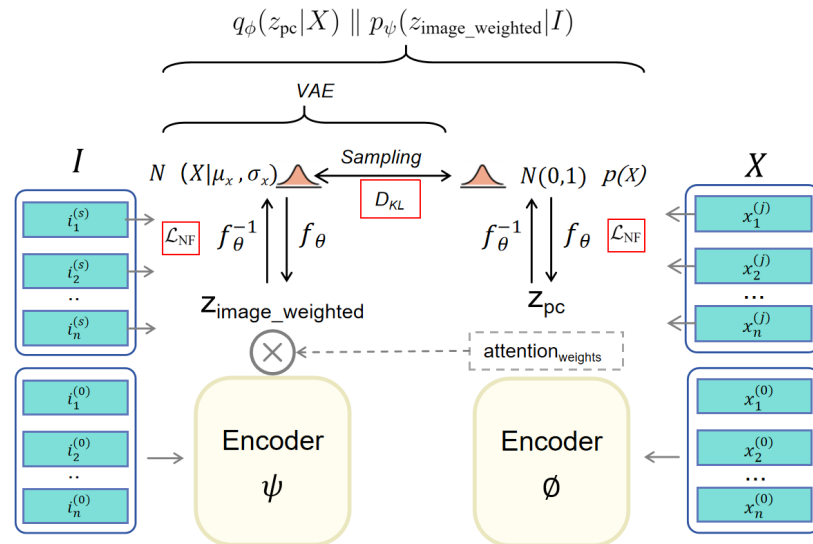
The generator consists of two domain-specific encoders and a decoder based on Normalizing Flows (NFs), as shown in Figure 3. In training, the image encoder  $\psi$  and point cloud encoder  $\phi$  work together to extract latent features from the input data. Specifically, the point cloud encoder  $\phi$ , based on PointNet++ [26], first processes the 3D point cloud data  $X$ . The point cloud encoder computes the mean  $\mu_X$  and standard deviation  $\sigma_X$  of the point cloud data, which are then used to generate a latent variable  $z_{pc}$ . This latent variable represents the high-dimensional abstract shape of the point cloud data. The goal of the point cloud encoder  $\phi$  is to learn the distribution of the latent variable  $q_\phi(z_{pc}|X)$  using a variational autoencoder (VAE), thereby approximating the true posterior distribution of the point cloud data  $p(z_{pc}|X)$ . The latent variable  $z_{pc}$  serves as prior knowledge for subsequent generative tasks, representing the underlying structure of the point cloud.

On the other hand, the image encoder  $\psi$ , which is based on ResNet [27], processes the image data associated with the point cloud. This encoder extracts rich features from the image, which are then fused with the point cloud features to enhance the generation process, especially in scenarios where visual data can provide useful priors for point cloud generation. The combined latent features from both encoders are subsequently passed to the decoder for point cloud generation.

$$z_{pc} = \mu_X + \sigma_X \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

where  $\mu_X$  and  $\sigma_X$  are the mean and standard deviation computed from the point cloud data  $X$ , and  $\epsilon$  is noise sampled from the standard Gaussian distribution  $\mathcal{N}(0, I)$ .





**Figure 3.** The point cloud generator in our model generates two important components: the Kullback–Leibler divergence term  $D_{KL}$ , which measures the difference between the approximate and true posterior distributions, and the normalizing flow model loss  $L_{NFs}$ , which helps optimize the latent space transformations. The different arrows indicate the flow of data through the generator, and the red box highlights the key components of the generator’s architecture.

Next, we incorporate image features to enhance the point cloud generation process. Before processing the image, we first divide the point cloud data into a 3D grid (i.e., voxels) and count the number of points within each voxel. This statistical information allows us to calculate the density  $\rho_i$  of each voxel, which is defined as the ratio of the number of points to the volume of the voxel. The formula is given by:

$$\rho_i = \frac{n_k}{V_k}$$

where  $\rho_i$  represents the density of the  $i$ th voxel,  $n_k$  is the number of points in voxel  $k$ , and  $V_k$  is the volume of that voxel. This formula quantifies the spatial sparsity or density of the point cloud and provides valuable information for subsequent feature fusion and generation processes, especially with respect to the spatial distribution of the point cloud.

After calculating the voxel density information, we use the image encoder  $\psi$  to extract the latent features  $Z_{image}$  from the image  $I$ . Next, each voxel density  $\rho_i$  is used to weight the image features  $Z_{image}$ , resulting in the weighted image latent features  $Z_{image\_weighted}$ . Specifically, the density of the point cloud determines the weighting factor for each image feature. To ensure that image regions corresponding to sparser point cloud areas receive more attention during feature fusion, we use the inverse of the point cloud density as the weighting factor:

$$\text{attention\_weights}_i = \frac{1}{\rho_i + \epsilon}$$

where  $\epsilon$  is a small constant to avoid division by zero. The weighted fusion is expressed as:

$$Z_{image\_weighted} = \sum_{i=1}^N \text{attention\_weights}_i \cdot Z_{image_i}$$

Here,  $Z_{image_i}$  denotes the  $i$ th element of the image latent features, and  $\text{attention\_weights}_i$  is the weighting factor computed from the point cloud density. The point cloud density  $\rho_i$  reflects the sparsity or density of each voxel. With this method, sparser point cloud regions are assigned more prominent weighting factors, ensuring that the image

features corresponding to these areas significantly influence the generation process. Finally, the weighted image latent features  $z_{\text{image\_weighted}}$  are mapped into the latent space along with the point cloud latent features  $z_{\text{pc}}$ , generating the image-conditioned latent distribution  $p_{\psi}(z_{\text{image\_weighted}}|I)$ .

To enhance the expressiveness of the generative model, we employ Normalizing Flows (NFs) to model the complex distribution of latent variables. This is achieved by using invertible transformations, such as affine coupling layers, to map a simple prior distribution (e.g., a standard Gaussian distribution  $\mathcal{N}(0, 1)$ ) to the complex point cloud distribution  $p(X)$ . The transformation process is performed by maximizing the log-likelihood of the data, specifically using Maximum Likelihood Estimation (MLE) to optimize the transformation function of the latent space, thereby improving the generative capability of the model. During the training of the normalizing flow, both the forward and inverse transformations are core operations. The inverse transformation (from  $p(X)$  to the standard Gaussian distribution  $\mathcal{N}(0, 1)$ ) is given by:

$$z_{\text{pc}} = f_{\theta}^{-1}(X), \quad f_{\theta}^{-1} : p(X) \rightarrow \mathcal{N}(0, 1)$$

This inverse transformation maps the complex point cloud distribution  $p(X)$  to the standard normal distribution  $\mathcal{N}(0, 1)$ , such that the latent variable  $z_{\text{pc}}$  follows a standard normal distribution. Conversely, the forward transformation (from the standard Gaussian distribution  $\mathcal{N}(0, 1)$  to the point cloud distribution  $p(X)$ ) is expressed as:

$$X = f_{\theta}(z_{\text{pc}}), \quad f_{\theta} : \mathcal{N}(0, 1) \rightarrow p(X)$$

This forward transformation allows the normalizing flow model to transform the standard normal distribution  $\mathcal{N}(0, 1)$  into the target point cloud distribution  $p(X)$ .

We model the entire generation process conditionally, i.e., generating point cloud data  $X$  given the latent variables  $z_{\text{pc}}$  and  $z_{\text{image\_weighted}}$ . To achieve this, we use a variational autoencoder (VAE) to infer the input data, calculating the mean and variance of the latent space and thereby defining the distribution of the latent variables. Specifically, the conditional likelihood of the point cloud data is given by:

$$p_{\theta}(X|z_{\text{image\_weighted}}) = \mathcal{N}(X|\mu_X, \sigma_X)$$

where  $\mu_X$  and  $\sigma_X$  are the conditional generation parameters computed based on the latent variable  $z_{\text{image\_weighted}}$ . During training, we jointly optimize the VAE and normalizing flows (NFs) models to maximize the Evidence Lower Bound (ELBO) while simultaneously incorporating the **Maximum likelihood estimation (MLE)** to adjust the model parameters.

For the VAE part, our objective is to maximize the ELBO, with the optimization objective expressed as:

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_{\phi}(z_{\text{pc}}|X)}[\ln p_{\theta}(X|z)] - D_{\text{KL}}[q_{\phi}(z_{\text{pc}}|X)||p_{\psi}(z_{\text{image\_weighted}}|I)]$$

Here, the first term is the reconstruction error  $\mathbb{E}_{q_{\phi}(z_{\text{pc}}|X)}[\ln p_{\theta}(X|z)]$ , also known as the generative error, which measures the difference between the generated point cloud  $X$  and the real point cloud. Minimizing this term makes the generated point cloud progressively closer to the real point cloud. The second term is the KL divergence  $D_{\text{KL}}[q_{\phi}(z_{\text{pc}}|X)||p_{\psi}(z_{\text{image\_weighted}}|I)]$ , which constrains the variational posterior  $q_{\phi}(z_{\text{pc}}|X)$  to be close to the prior distribution  $p_{\psi}(z_{\text{image\_weighted}}|I)$ . Minimizing this term encourages the posterior distribution to approximate the true posterior, thereby enhancing the generative capability of the model.

During the flow model training, we optimize the normalizing flow transformation function  $f_\theta$  by maximizing the log-likelihood. The objective function is given by:

$$\mathcal{L}_{\text{NFs}} = \mathbb{E}_{q_\phi(z_{\text{pc}}|X)}[\ln p_\theta(X|z)]$$

Here,  $\mathbb{E}_{q_\phi(z_{\text{pc}}|X)}[\ln p_\theta(X|z)]$  represents the reconstruction error, which measures the difference between the generated point cloud  $X$  and the real point cloud through the latent variable  $z_{\text{pc}}$ . By maximizing this objective function, the normalizing flow learns the invertible transformation from the standard normal distribution (or other prior distributions) to the target distribution (i.e., the point cloud data distribution). The final joint loss function for the entire model combines the VAE reconstruction error, the KL divergence, and the normalizing flow log-likelihood objective:

$$\mathcal{L} = \sum_X \mathbb{E}_{q_\phi(z_{\text{pc}}|X)}[\ln p_\theta(X|z)] - D_{\text{KL}}[q_\phi(z_{\text{pc}}|X) || p_\psi(z_{\text{image\_weighted}}|I)]$$

The first term  $\sum_X \mathbb{E}_{q_\phi(z_{\text{pc}}|X)}[\ln p_\theta(X|z)]$  represents the goal of fitting the real point cloud through the latent variable  $z_{\text{pc}}$ , while the second term is the KL divergence, which constrains the variational posterior distribution. Without the generative model, the Maximum Likelihood Estimation (MLE) directly maximizes the log-likelihood during training to optimize the generative model.

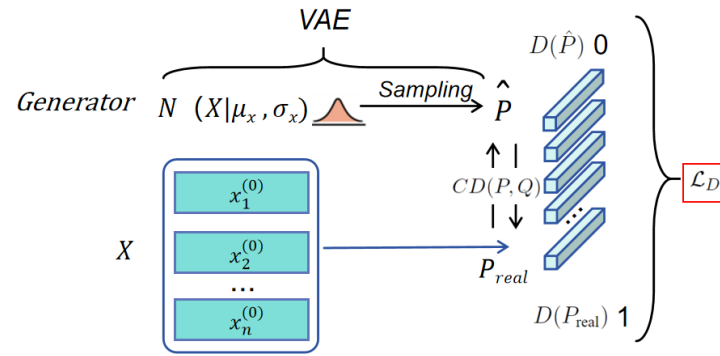
By maximizing the log-likelihood and minimizing the KL divergence through MLE, the model can generate point clouds that closely match the real data distribution while preserving the structural information of the latent space. This process effectively improves the model's generative capability.

### 3.3. Points Cloud Discriminator

After the generator's inference, we successfully simulated the distribution of point clouds in the real world and generated highly realistic sample data. To further enhance the sample quality and train the generator more effectively, we adopted the generative adversarial network (GAN) framework, which consists of a generator and a discriminator. In the discriminator, we aimed to distinguish between the fake point clouds generated by the generator and the real point clouds, as shown in Figure 4. To achieve this, we chose the Chamfer Distance (CD) as the metric for measuring the similarity between 3D point clouds. Unlike traditional pixel- or voxel-based metrics, the Chamfer Distance can accurately measure the differences between point sets, thereby providing a more precise evaluation of the similarity between generated and real point clouds. The Chamfer Distance (CD) is defined in Equation (1):

$$CD(P, Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|p - q\|^2 + \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \|p - q\|^2 \quad (1)$$

where  $P$  and  $Q$  are two point clouds (one generated, one real),  $\|p - q\|$  is the Euclidean distance between a point  $p \in P$  and a point  $q \in Q$ , and  $|P|$  and  $|Q|$  represent the number of points in each point cloud. The goal of the Chamfer Distance is to penalize the generator when the generated point clouds significantly differ from the real data, prompting it to generate more realistic point clouds over time.



**Figure 4.** The point cloud discriminator utilizes the loss function  $\mathcal{L}_D$ , which plays a critical role in adversarial training by quantifying the discriminator's success in correctly identifying real point cloud data as real and generated point cloud data as fake. The minimization of  $\mathcal{L}_D$  helps the model improve the realism of the generated point clouds. The red box highlights the key part of the discriminator's architecture.

To evaluate the effectiveness of the discriminator, we designed a loss function to promote the adversarial behavior between the generator and the discriminator. In the GAN framework, the generator's objective was to minimize the Chamfer Distance, thereby reducing the difference between the generated and real data. On the other hand, the discriminator aimed to maximize that difference. Through this adversarial process, both the generator and discriminator improved iteratively. In our experiments, we incorporated a feature extractor to extract higher-level features from the point clouds, which were then fed into the discriminator. This ensured that the discriminator considered not only the point coordinates but also the higher-level semantic features, thereby enhancing its ability to evaluate the quality of generated data.

To further evaluate the effectiveness of the discriminator, we used a loss function to promote the adversarial behavior between the generator and the discriminator. In the GAN framework, the generator aimed to deceive the discriminator by generating the most realistic fake point clouds, while the discriminator attempted to distinguish between real and fake data. The generator's objective was to minimize the Chamfer Distance and reduce the difference between the generated and real point clouds, while the discriminator tried to maximize that difference. This adversarial training process encouraged the generator to improve its generation capabilities continuously.

The optimization process for training the discriminator can be represented as:

$$\mathcal{L}_D = -\mathbb{E}_{P_{\text{real}}}[\log D(P_{\text{real}})] - \mathbb{E}_{\hat{P}}[\log(1 - D(\hat{P}))] \quad (2)$$

where  $P_{\text{real}}$  is the real point cloud sampled from the real dataset,  $\hat{P}$  is the fake point cloud generated by the generator, and  $D(P)$  is the discriminator's probability  $P$  of being a real point cloud. The discriminator's loss function  $\mathcal{L}_D$  aims to maximize the probability  $D(P_{\text{real}})$  of real point clouds being classified as real while minimizing the probability  $D(\hat{P})$  of fake point clouds being classified as real. By minimizing this loss function, the discriminator improves its ability to distinguish between generated and real point clouds, thus enhancing the quality of the generator's generated point clouds.

### 3.4. Glossary

This section provides a list of symbols used in this paper, along with their corresponding definitions (Table 1).

**Table 1.** Symbols and definitions.

Symbol	Definition
$I$	Weighted image feature distribution
$P$	Point cloud feature distribution
VAE	Variational inference
NFs	Normalizing flow model
GAN	Generative adversarial network
$\phi$	Point cloud encoder
$\psi$	Image encoder
$\rho_i$	Voxel density
$i$	Voxel
<i>point_cloud_densities</i>	Point cloud data density
<i>image_data</i>	Image data
<i>fused_feature</i>	Fused feature
<i>attention_weights</i>	Attention weights
$n_k$	Number of points in voxel $k$
$V_k$	Volume of voxel $k$
$N(0, 1)$	Gaussian distribution
$F$	Affine coupling layer
$z$	Latent vector
$\theta$	Forward transformation of affine coupling layer
$\theta^{-1}$	Inverse transformation of affine coupling layer
ELBO	Maximizing the negative Evidence Lower Bound (ELBO)
KL	Kullback–Leibler divergence (KL divergence)
MLE	Maximum likelihood estimation
CD	Chamfer distance
$q_\phi(z_{pc} X)$	Approximate posterior distribution of point cloud
$p_\psi(z_{\text{image\_weighted}} I)$	Prior distribution conditioned on the image data
$P_{\text{real}}$	Real point cloud data
$\hat{P}$	Generated pseudo-point cloud data
$G(z)$	Output of the generator
$D_{KL}$	Kullback–Leibler divergence loss
$L_{VAE}$	Variational inference loss
$L_{NFs}$	Normalizing flow model loss
$\mathcal{L}_D$	Discriminator loss
$G$	Generator
$D$	Discriminator
$\mathcal{L}$	Generator loss

## 4. Experiments

The general training objectives and algorithms presented in the previous section lay the foundation for formulating specific point cloud tasks. Next, we adjusted the training objectives for the point cloud generation capability and the feasibility of generation in real scenarios, respectively. Regarding section organization, we first introduce the experimental environment; the datasets used the evaluation metrics and other relevant information about the experimental setup. Subsequently, we demonstrate the model’s generation capabilities on multiple datasets and prove its advantages and effectiveness in different tasks through quantitative and qualitative analyses. Finally, we discuss the limitations of this method and propose further optimization directions.

### 4.1. Experimental Setup

This section outlines the setup and experimental procedures used to evaluate the performance of our proposed model. Our experiments were designed to assess the model’s feasibility and effectiveness in synthetic and real-world scenarios. The primary goals of the

experiments were to validate the model's point cloud generation capabilities, compare it against benchmark models, and analyze its performance using various evaluation metrics.

The experiments were conducted on a machine equipped with a single Nvidia 3090 GPU (NVIDIA Corporation, Santa Clara, CA, USA), running the CUDA 12.1 driver and using the PyTorch deep learning framework. The GPU had 24 GB of video memory, which was sufficient for processing the large datasets used in the experiments. The model's point cloud generation capabilities were first tested on the synthetic ShapeNet dataset, followed by real-world data from the nuScenes dataset. Both datasets were selected for their relevance to point cloud generation tasks in different settings: ShapeNet for controlled, synthetic data and nuScenes for complex, real-world data.

#### 4.1.1. Dataset

We evaluated the model's performance using two distinct datasets: the ShapeNet-Core.v1 dataset and the nuScenes dataset. These datasets were selected for their complementary nature: ShapeNet provides a controlled environment, while nuScenes offers real-world complexity.

ShapeNetCore.v1 is a well-established synthetic dataset containing an extensive collection of 3D object models across 55 categories and corresponding 2D images. This dataset was chosen because it offered complete and aligned image–point cloud pairs, which were crucial for training and evaluating the model. It also provided a robust foundation for comparing our model's performance against existing point cloud generation methods.

nuScenes is a real-world, multimodal dataset used extensively in autonomous driving research. It contains data collected from multiple cities and includes synchronized LiDAR, camera, radar, and IMU sensors. This dataset was particularly valuable for evaluating our model in real-world scenarios, as it featured complex urban environments with varying levels of point cloud sparsity. The nuScenes dataset was ideal for assessing the model's ability to handle the challenges of outdoor, real-time point cloud generation.

#### 4.1.2. Evaluation Metrics

To assess the performance of our model, we employed several quantitative and qualitative metrics to evaluate the similarity between generated and real point clouds. The metrics used were the Chamfer Distance (CD), Earth Mover's Distance (EMD), and the F1 score. Each of these metrics focused on different aspects of the model's performance.

- Chamfer Distance (CD): this metric measures the average squared Euclidean distance between points in two sets of point clouds, measuring spatial similarity between the generated and real data.

$$D_{CD}(P, Q) = \frac{1}{N_P} \sum_{p \in P} \min_{q \in Q} \|p - q\|^2 + \frac{1}{N_Q} \sum_{q \in Q} \min_{p \in P} \|q - p\|^2$$

- Earth Mover's Distance (EMD): the EMD quantifies the minimum amount of work required to transform one distribution of points into another, effectively capturing the geometric and topological differences between two point clouds.

$$D_{EMD}(P, Q) = \min_{f: P \rightarrow Q} \sum_{p \in P} \|p - f(p)\|$$

- **F1 Score:** The F1 score is the harmonic mean of precision and recall, and it evaluates the accuracy of object recognition and segmentation in point cloud data. This metric is critical when dealing with sparse and complex point cloud data.

$$D_{F1}(P, Q) = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where precision and recall are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

Here, TP represents true positives, FP denotes false positives, and FN refers to false negatives.

These metrics helped evaluate the quality and fidelity of the generated point clouds, ensuring that the model accurately reconstructed object shapes, recovered missing details, and aligned well with the real data.

#### 4.1.3. Training Process

During the training phase, **when trained on ShapeNetCore.v1 the model input contains an image and a corresponding point cloud from the same object category. However, when training on nuScenes, each training sample includes eight image samples and one point cloud sample.** The image features were extracted using ResNet [28], while the point cloud features were extracted using PointNet++ [26]. These features were concatenated and passed into the model as input. The Adam optimizer was used to adjust the learning rate and minimize the loss function, facilitating effective learning of the point cloud generation task. When real-world data were used, only 2500 sampling points were selected for training to reduce computational complexity. We visualized the generated point clouds for evaluation to provide qualitative insights into the model's performance.

#### 4.2. Evaluation

This section comprehensively evaluates the model's effectiveness through a series of experiments. Given the current limited research on point cloud generation in sparse regions and real-world scenarios, we selected two representative and widely used techniques in the field as benchmark methods: the flow-based model PointFlow [18] and diffusion probability model [29], which is renowned for its strong generative capabilities. These methods have demonstrated outstanding representativeness and effectiveness in point cloud generation tasks. Moreover, many existing related works can be regarded as their variants, often accompanied by higher model complexity. This complexity poses higher demands on computing resources when generating point clouds in extreme environments or under resource-constrained conditions. Therefore, we chose representative models with lower computing resource requirements to verify the feasibility and advantages of our method.

Firstly, we evaluated the performance of our model on the ShapeNet synthetic dataset and compared it with existing point cloud generation methods, specifically including the flow-based method PointFlow [18] and the GAN-based diffusion model Diffusion [29]. This comparison aimed to assess our model's quality and efficiency in generating realistic and high-quality point clouds. The experimental results showed that our model outperformed the benchmark methods across multiple key evaluation metrics, such as the Chamfer Distance (CD), Earth Mover's Distance (EMD), and F1 score. These metrics were used to evaluate the spatial alignment, geometric accuracy, and recognition capability of the generated point clouds, further confirming the superiority of our approach.

Additionally, we performed classification on the generated point clouds using PointNet++ [26] to further assess the feasibility of our model in real-world scenarios. After generating the point clouds, we applied PointNet++ to classify them and compared the classification performance with that of the original point cloud data. The results demonstrated that the classification accuracy improved by 3% after integrating the generated point clouds. This indicated that our model not only generated high-quality point clouds but also enhanced the utility of the generated data in downstream tasks such as classification.

Next, we evaluated our model’s feasibility in generating point clouds for real-world scenarios by conducting experiments on the nuScenes dataset, which contains real-world data from outdoor, unknown environments. Unlike the ShapeNet dataset, which is synthetic, the nuScenes dataset includes data from autonomous driving, providing a more complex and diverse set of scenarios for point cloud generation. The input format for the nuScenes dataset consisted of eight images and a single point cloud file, which differed from the single image and point cloud pairs used in the ShapeNet dataset. This allowed us to assess our model’s ability to handle more complex real-world data, and the results further demonstrated the effectiveness of our model in generating high-quality point clouds in real-world environments.

Furthermore, we conducted an ablation study on the nuScenes dataset to explore the impact of the point cloud density-based attention mechanism in our model. That mechanism helped to enhance the model’s performance in generating sparse point cloud data in edge regions. By visualizing the generated point clouds and comparing them with baseline models such as PointFlow [18] and Diffusion [29], we observed that our model was particularly effective at improving the representation of distant objects and edge features. The ablation experiments’ results further validated the attention mechanism’s importance in generating more accurate and detailed point cloud data, particularly in sparse regions.

#### 4.2.1. Point Cloud Generator

The experimental comparison of our model’s performance with baseline methods, such as PointFlow and Diffusion, demonstrated the effectiveness of our approach in point cloud generation. Specifically, the results showed that our model significantly outperformed the baseline methods regarding the Earth Mover’s Distance (EMD), indicating better overall alignment and accuracy of the generated point clouds. Furthermore, our model also achieved superior performance in the Chamfer Distance (CD) and F1 score metrics, which assessed the spatial proximity and object recognition capability of the generated point clouds.

As shown in Table 2, these experiments demonstrated that our model had outstanding capabilities in the point cloud generation task and significantly outperformed the benchmark methods (such as PointFlow and **Diffusion Probabilistic Model**).

**Table 2.** Evaluation on the ShapeNet simulation dataset.

ShapeNet Dataset	CD $\times 10^3$	EMD $\times 10^2$	F1
PointFlow	10.22	6.58	-
Diffusion Probabilistic Model	3.25	10.21	34.06
Ours	2.32	8.81	79.81

In this experiment, we evaluated the performance of our model by performing a point cloud classification task on the generated point cloud data. Specifically, we fused the generated point clouds with the original point cloud data and then conducted the classification task to assess whether this fusion could improve classification performance. The results showed that, compared to the baseline classification performance using only the original point cloud data, the classification accuracy increased by 3% after integrating



the generated point clouds. This improvement indicated that the generated point cloud data enhanced classification performance, particularly when the original point cloud data were sparse or incomplete. As shown in Table 3, the quantitative results clearly demonstrated the positive impact of the generated point clouds on classification accuracy, further validating the effectiveness of our model in improving point cloud classification in practical applications.

**Table 3.** Classification accuracy of the generated results.

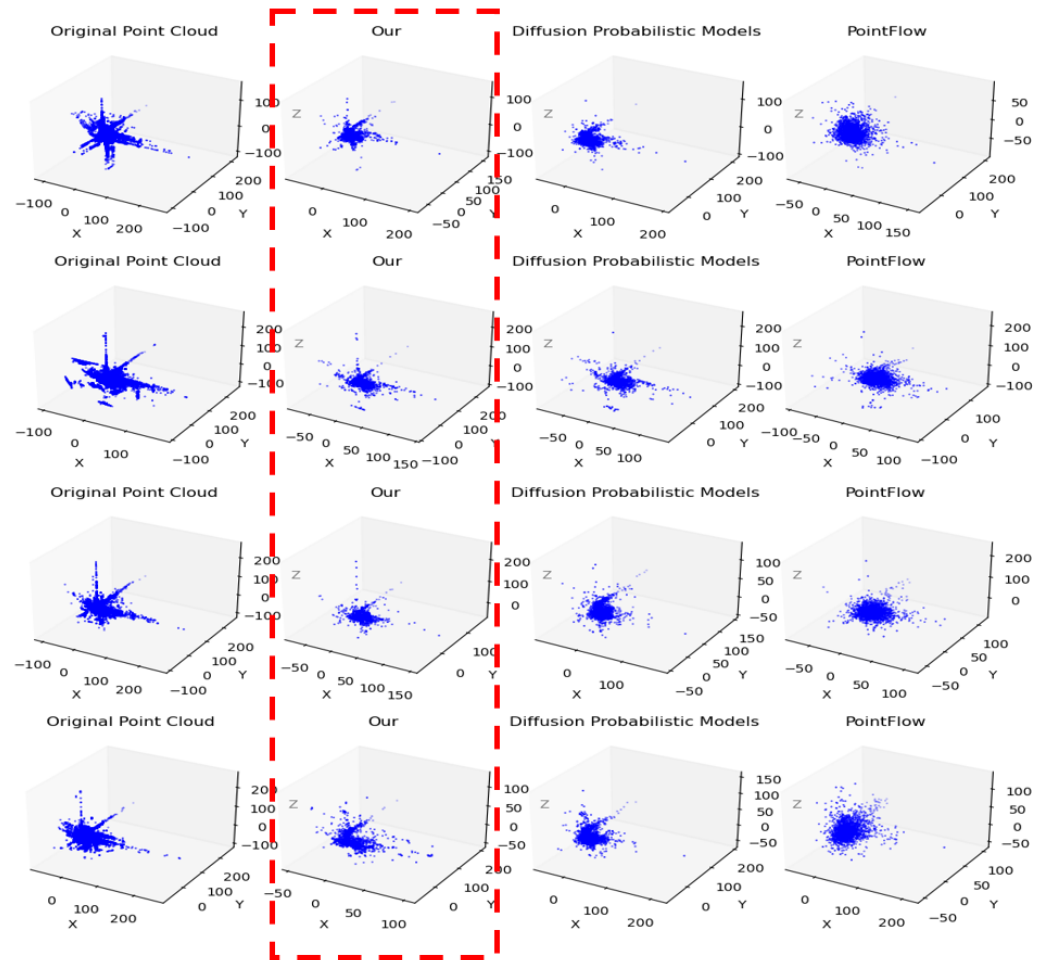
Classification	Accuracy	avg_accuracy	mIOU	avg_mIOU
Pointnet++	0.94447	0.87544	0.82647	0.85540
Ours	0.96194	0.90984	0.86556	0.88709

This result highlights the advantages of our generative model in terms of data completeness and feature representation and its ability to supplement sparse regions in the point clouds, thereby enhancing the classification model’s understanding and recognition of the data. While the 3% improvement may seem modest, it is significant in scenarios involving complex environments or incomplete data, such as autonomous driving, air-ground collaboration, and extreme conditions. The generated point clouds effectively fill in gaps in the original data, improving system stability and robustness. Additionally, this experiment demonstrates that our model improves point cloud classification accuracy and supports subsequent tasks, such as 3D map construction. In practical applications, the generated point clouds contribute to data robustness and completeness, providing more reliable inputs for high-precision task execution. Overall, the results validate the superiority of our model in generating and enhancing 3D point cloud data, offering strong evidence for its broad applicability in real-world scenarios.

#### 4.2.2. Real-World Point Cloud Generation in Outdoor Unknown Environments

In this experiment, we evaluated the feasibility of our model using the nuScenes real-world dataset to test its ability to generate point clouds in unknown outdoor environments. We compared our method with existing generation methods, particularly PointFlow [18] and Diffusion Probabilistic models [29], both of which are designed for real-scene point cloud generation. Additionally, we provide visual comparisons that clearly demonstrate the performance of our method in these challenging scenarios, as shown in Figure 5. Regarding the generation quality, our method produced more apparent shape features with more uniform point cloud distribution and higher resolution. It maintained higher geometric accuracy, especially when capturing complex shapes and structures in intricate outdoor environments. Moreover, our method exhibited strong robustness across various environmental conditions, consistently generating stable point clouds. Importantly, our model excelled at handling sparse areas, effectively filling in regions that other methods may overlook. These advantages enable our model to generate more reliable point clouds for practical applications, particularly in real-world scenarios.

In contrast, PointFlow demonstrated limited effectiveness in generating real-world scenes, with results indistinguishable from noise. Although PointFlow achieved relatively good results on synthetic datasets, it failed to capture clear shape features in complex real environments, leading to generated results that lacked clarity and structure. Consequently, PointFlow’s performance in generating real-world point clouds was inferior to our model, especially in detail recovery and handling sparse areas.



**Figure 5.** Visual results of our method and the benchmark method on nuScenes when comparing the generation effects of 3D point cloud data. The red box highlights the key area of interest, showing the visualization results of our model. This part illustrates the model's performance, specifically emphasizing the key features and improvements of the generated 3D point cloud data compared to the benchmark method.

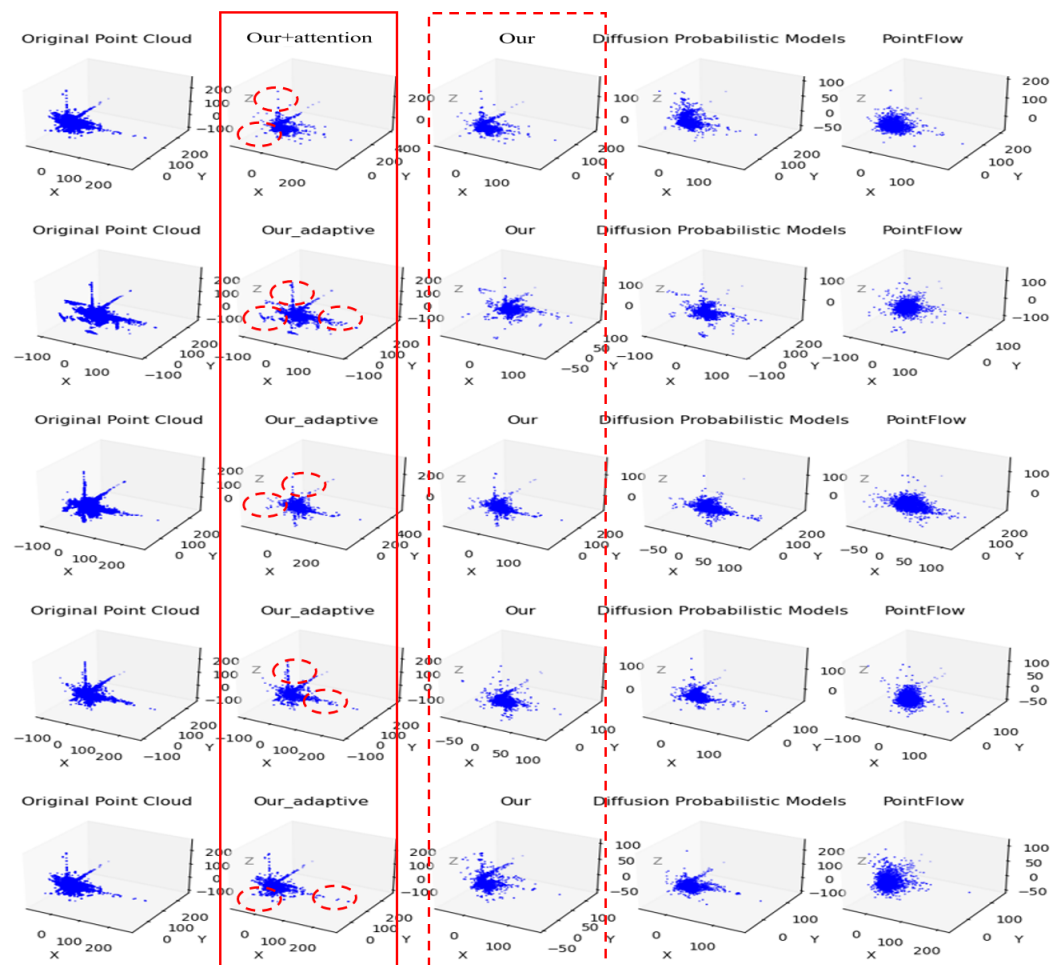
On the other hand, while the Diffusion Probabilistic Model could initially present some shape features when generating point clouds, they still produced a significant amount of noise. This indicates that under the same conditions, our model converged faster and generated point clouds more efficiently. Despite the Diffusion Probabilistic Model's ability to capture basic shape contours in some cases, the noise problem degraded the quality of the generated point clouds and limited their performance in complex scenarios. In comparison, our method significantly outperformed the Diffusion Probabilistic Model in generation efficiency, producing high-quality point clouds more quickly while effectively reducing noise and improving geometric accuracy and structural clarity. Through comparisons with PointFlow and Diffusion Probabilistic Model, our model demonstrated significant advantages in point cloud generation quality, efficiency, and robustness, particularly in complex outdoor unknown environments.

#### 4.2.3. Ablation Study

In this ablation experiment, we investigated the impact of the point cloud density-based attention module by comparing the model's performance with and without the attention mechanism. We conducted experiments on the nuScenes dataset, where the model was fed with trained point cloud and image data. We then evaluated the model with

and without the attention module. The results demonstrated a noticeable degradation in point cloud quality and a significant change in the generation process when the attention module was removed.

The experimental results showed that the attention module was crucial for generating point cloud data in sparse regions, particularly at the edges. The model struggled to accurately represent distant objects without the attention mechanism, resulting in point clouds with blurred geometric details. On the other hand, with the attention module enabled, the model could better represent distant objects, and the generated point clouds exhibited notable improvements in geometric fidelity and shape features, especially in complex scenes. As shown in Figure 6, the generated point clouds were more precise and more accurate in terms of both geometric details and shape representation.



**Figure 6.** Results of ablation experiments conducted on the nuScenes dataset. To better demonstrate the role of the attention module in generating point cloud shape features, we visualized the inference results of these ablation experiments and compared them with the baseline model. Among them, “Ours + attention” represents the complete model structure, and “Ours” represents the model without the attention structure. The circled regions highlight areas where the point cloud’s feature representation ability significantly declines after removing the attention mechanism.

Furthermore, we compared the results of the ablation model with baseline models, such as PointFlow [18] and Diffusion Probabilistic Models [29]. The results demonstrated that the model with the attention mechanism outperformed the baseline models, particularly in capturing distant objects and enhancing geometric features. This validated the effectiveness of the attention module in improving model performance, particularly in generating high-quality point clouds in complex real-world scenarios.

These experiments underscore the critical role of the point cloud density-based attention mechanism in generating point clouds for sparse regions, enhancing geometric features, and improving model robustness. The findings provide strong evidence supporting the effectiveness of our method in practical applications.

## 5. Discussion

I-PAttnGAN performed exceptionally well in point cloud generation, especially in sparse areas, capable of generating highly accurate and complete 3D point clouds suitable for practical scenarios. The combination of image-assisted input and the attention mechanism significantly enhanced the model's generation ability, making it excel in reconstructing missing regions. Notably, in the case of point clouds with sparse edges, this model excelled in preserving shape features, and the joint use of image data and the attention mechanism enabled the model to intelligently focus on key areas when dealing with sparse data, thereby improving the overall quality of the generated point clouds.

Compared with existing point cloud generation methods (such as those based on flow and diffusion models), I-PAttnGAN demonstrated significant advantages in both generation quality and training efficiency. Its ability to utilize the attention mechanism enabled it to prioritize the processing of important regions in the point cloud, thereby achieving a more accurate and efficient reconstruction process.

However, the method adopted by I-PAttnGAN, namely, the point cloud attention module based on key regions of the image, is highly dependent on the quality of the input image. The model's performance may significantly decline when the input image is of poor quality, occluded, or contains occlusions. In such cases, the generated point clouds may have incomplete or inaccurate structures, which could adversely affect downstream tasks such as map construction. This issue is particularly prominent in real-world applications, as point cloud data often come from various sources, and the related images may not always meet the quality standards required for achieving optimal performance.

Furthermore, when confronted with noisy data in complex and unknown outdoor scenes, the model's adaptability will decline. A large amount of noise in point cloud data further increases the difficulty, especially when the environmental background and scene features change significantly. Therefore, when applied to diverse real-world datasets, the model's performance may drop, limiting its ability to handle the inherent variations in different scene types.

To address these challenges, future research efforts can focus on enhancing the model's robustness to low-quality or noisy input data, for instance, by integrating multi-perspective data, depth maps, or advanced denoising techniques. Additionally, exploring methods to improve the model's adaptability to different scenarios, such as through domain adaptation strategies or using more general training data, may help extend the applicability of I-PAttnGAN to a broader range of practical applications.

## 6. Conclusions

The I-PAttnGAN method proposed in this study effectively enhanced the representation ability of long-distance point clouds in real scenarios by introducing image feature guidance, compensating for their sparsity. This helped solve related problems in extreme environments, such as point cloud missing and demonstrated outstanding performance in challenging tasks like lake surface reconstruction. Additionally, this method lays a solid foundation for special functions that rely on low-overlap areas for registration in air-ground collaborative scenarios.

Future work will focus on low-overlap point cloud registration, enhancing registration accuracy and robustness, and supporting a broader range of practical applications.

**Author Contributions:** Conceptualization, W.L. and Y.C.; methodology, W.L., Y.C. and Q.F.; software, W.L. and Q.F.; validation, Y.C.; formal analysis, Y.C.; investigation, W.L. and Y.C.; resources, Y.C., B.G. and Z.Y.; data curation, W.L. and Y.C.; writing—original draft preparation, W.L., Y.C. and Q.F.; writing—review and editing, Y.C., M.Y., B.G. and Z.Y.; visualization, W.L.; supervision, Y.C., B.G. and Z.Y.; project administration, Y.C.; funding acquisition, Y.C., B.G. and Z.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China under grant no. 62202379, The Fundamental Research Funds for the Central Universities under grant no. G2022KY05102.

**Data Availability Statement:** The data supporting the reported results can be found in the nuScenes dataset, available at <https://www.nuscenes.org/>, and the ShapeNetCore.v1 dataset, available at <https://shapenet.org/>.

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

## References

1. Zhang, Y.; Carballo, A.; Yang, H.; Takeda, K. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *Isprs J. Photogramm. Remote. Sens.* **2023**, *196*, 146–177. [[CrossRef](#)]
2. Gupta, A.; Ingle, A.; Velten, A.; Gupta, M. Photon-Flooded Single-Photon 3D Cameras. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 6763–6772. [[CrossRef](#)]
3. Yoneda, K.; Suganuma, N.; Yanase, R.; Aldibaja, M. Automated driving recognition technologies for adverse weather conditions. *Latss Res.* **2019**, *43*, 253–262. [[CrossRef](#)]
4. Li, C.L.; Zaheer, M.; Zhang, Y.; Poczos, B.; Salakhutdinov, R. Point Cloud GAN. *arXiv* **2018**, arXiv:1810.05795.
5. 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/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
6. Wei, Y.; Vosselman, G.; Yang, M.Y. Flow-Based GAN for 3D Point Cloud Generation from a Single Image. *arXiv* **2022**, arXiv:2210.04072.
7. Ko, J.; Cho, K.; Choi, D.; Ryoo, K.; Kim, S. 3D GAN Inversion With Pose Optimization. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 2967–2976.
8. Chen, J.; Li, G.; Zhang, R.; Li, T.H.; Gao, W. Pointvae: Invertible Variational Autoencoder Framework for 3D Point Cloud Generation. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 3216–3220. [[CrossRef](#)]
9. Razavi, A.; van den Oord, A.; Vinyals, O. Generating Diverse High-Fidelity Images with VQ-VAE-2. *arXiv* **2019**, arXiv:1906.00446.
10. Tomczak, J.; Welling, M. VAE with a VampPrior. In Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, Playa Blanca, Lanzarote, 9–11 April 2018; Storkey, A., Perez-Cruz, F., Eds.; Proceedings of Machine Learning Research; PMLR: Birmingham, UK, 2018; Volume 84, pp. 1214–1223.
11. Anvekar, T.; Tabib, R.A.; Hegde, D.; Mudengudi, U. VG-VAE: A Venatus Geometry Point-Cloud Variational Auto-Encoder. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA, 19–20 June 2022; pp. 2977–2984. [[CrossRef](#)]
12. Yuan, W.; Khot, T.; Held, D.; Mertz, C.; Hebert, M. PCN: Point Completion Network. *arXiv* **2019**, arXiv:1808.00671.
13. van den Oord, A.; Kalchbrenner, N.; Vinyals, O.; Espeholt, L.; Graves, A.; Kavukcuoglu, K. Conditional Image Generation with PixelCNN Decoders. *arXiv* **2016**, arXiv:1606.05328.
14. Xue, X.; Jia, J. Visual Tracking by Gated PixelCNN Model. In Proceedings of the CSAI'19, 2019 3rd International Conference on Computer Science and Artificial Intelligence, Normal, IL, USA, 6–8 December 2019; pp. 165–170. [[CrossRef](#)]
15. Dinh, L.; Sohl-Dickstein, J.; Bengio, S. Density estimation using Real NVP. *arXiv* **2017**, arXiv:1605.08803.
16. Kingma, D.P.; Dhariwal, P. Glow: Generative Flow with Invertible 1x1 Convolutions. *arXiv* **2018**, arXiv:1807.03039.
17. Sun, Y.; Wang, Y.; Liu, Z.; Siegel, J.E.; Sarma, S.E. PointGrow: Autoregressively Learned Point Cloud Generation with Self-Attention. *arXiv* **2019**, arXiv:1810.05591.
18. 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/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
19. Klovov, R.; Boyer, E.; Verbeek, J. Discrete Point Flow Networks for Efficient Point Cloud Generation. *arXiv* **2020**, arXiv:2007.10170.

20. Xie, H.; Yao, H.; Sun, X.; Zhou, S.; Zhang, S. Pix2Vox: Context-Aware 3D Reconstruction From Single and Multi-View Images. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
21. Xu, C.; Yang, S.; Galanti, T.; Wu, B.; Yue, X.; Zhai, B.; Zhan, W.; Vajda, P.; Keutzer, K.; Tomizuka, M. Image2Point: 3D Point-Cloud Understanding with 2D Image Pretrained Models. *arXiv* **2022**, arXiv:2106.04180.
22. Wang, Z.; Rao, Y.; Yu, X.; Zhou, J.; Lu, J. Point-to-Pixel Prompting for Point Cloud Analysis With Pre-Trained Image Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 4381–4397. [[CrossRef](#)] [[PubMed](#)]
23. Mikuni, V.; Nachman, B.; Pettee, M. Fast point cloud generation with diffusion models in high energy physics. *Phys. Rev. D* **2023**, *108*, 036025. [[CrossRef](#)]
24. Yang, L.; Zhang, D.; Karniadakis, G.E. Physics-Informed Generative Adversarial Networks for Stochastic Differential Equations. *arXiv* **2018**, arXiv:1811.02033. [[CrossRef](#)]
25. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved Training of Wasserstein GANs. *arXiv* **2017**, arXiv:1704.00028. [[CrossRef](#)].
26. 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, NIPS'17, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
28. Targ, S.; Almeida, D.; Lyman, K. Resnet in Resnet: Generalizing Residual Architectures. *arXiv* **2016**, arXiv:1603.08029.
29. Luo, S.; Hu, W. Diffusion Probabilistic Models for 3D Point Cloud Generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 2837–2845.

**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.