LPCG: A Self-conditional Architecture for Labeled Point Cloud Generation

Dongshuo Huang[1]*, Xiaoshui Huang[2,3]*, Chengdong Zhang[2], Yilei Shi[1]†

[1]School of Software, Northwestern Polytechnical University
[2]School of Public Health, Shanghai Jiao Tong University School of Medicine
[3]School of Computer and Artificial Intelligence, Huaihua University
huangdongshuo@mail.nwpu.edu.cn, yilei_shi@nwpu.edu.cn, huangxiaoshui@163.com, zhangcd@sjtu.edu.cn

Abstract

Recently, there has been considerable exploration of methods for generating 3D point clouds, which is crucial for numerous 3D vision applications. Though conditional generation methods show promising performance, it depends on the additional paired label. On the other hand, unconditional generation methods usually fail to annotate the generated 3D point cloud. In this paper, we introduce a novel self-conditional architecture that trains on unlabeled data and then generates high-quality labeled 3D point clouds. Specifically, we design a module to extract geometry and view features, and then use a feature fusion module to integrate them as a substitute for label embedding in conditional point cloud generation. Then the point cloud generator is trained using the fused features. LPCG also harnesses CLIP to handle the view features of point clouds for generating label information. Besides, we train two feature diffusion modules to capture the essence of multimodal features and obtain diverse fused features for use as conditions in generating point clouds. Experiments on the ShapeNet dataset demonstrate that LPCG achieves state-of-the-art performance for single class generation. Our experimental results show that the accuracy of our generated label annotations reaches around 97.44% for a two-class generation task.

Introduction

Point cloud data finds extensive applications across diverse fields like autonomous driving, robotics, augmented reality, and virtual reality. Nevertheless, acquiring point cloud data necessitates researchers to transport a range of sensors, gather diverse data, and subsequently annotate the point cloud data with corresponding labels, demanding substantial human effort. Thus, exploration of better generative models (Wu et al. 2024; Zhou et al. 2023) for obtaining or augmenting high-fidelity 3D point cloud has become an active research topic in recent years. We believe that an ideal 3D point cloud generation model should have the following functionalities: 1) capable of generating high-quality point clouds based on merely existing 3D data; 2) allowing users to utilize a single model to learn the representation of various categories, and then generate corresponding point cloud data; 3) providing high-quality label of the generated point cloud for future use. Though massive explorations have been carried out, it is still challenging to build a model to meet all above criteria.

Generally speaking, existing research on 3D point cloud generation mainly falls into two categories. The first category is conditional generation (Fan, Su, and Guibas 2017; Kurenkov et al. 2018), which guides point cloud generation by specific conditions. Though conditional methods generally achieve a better performance, they usually require additional conditions to be attached to the point cloud training datasets. The second category is unconditional generation (Achlioptas et al. 2018; Yang et al. 2019). Instead of relying on the labels, unconditional methods leverage the internal features of the point cloud for modeling and generation. Unfortunately, when dealing with multi-category generation tasks, these methods can only generate complete point clouds and cannot distinguish the category to which the point cloud belongs. Besides, so far the generation quality is also dominated by conditional generation methods.

In this paper, we are inspired by RCG (Li, Katabi, and He 2023) model and consider the above two-category methods, and then propose a self-conditional 3D point cloud generation architecture, which is capable of generating point cloud with annotation, as shown in Figure 1.

Our approach harnesses the multimodal fused features within raw unlabeled data to train the generative model, circumventing the labor-intensive task of data annotation. We integrate the geometric feature and view features of the point cloud data to get the fused features. Our method not only elevates the quality of the produced point clouds but also employs CLIP to process view features, thereby furnishing annotated labels for the generated point clouds. Specifically, we first use a pre-trained self-supervised point cloud model and CLIP to extract geometric and view features of the point cloud, and then integrate them by feature fusion module for a substitute for label embedding. Secondly, we train the feature diffusion modules by adding and removing noise on the previously extracted feature to provide diverse conditions for the following point cloud generator. Finally, we train the point cloud generator conditioned on the extracted fused features and provide two sampling methods during inference: one based on feature diffusion and the other on the training dataset. Simultaneously, we use CLIP to handle the sampled

*These authors contributed equally.
†Corresponding Author

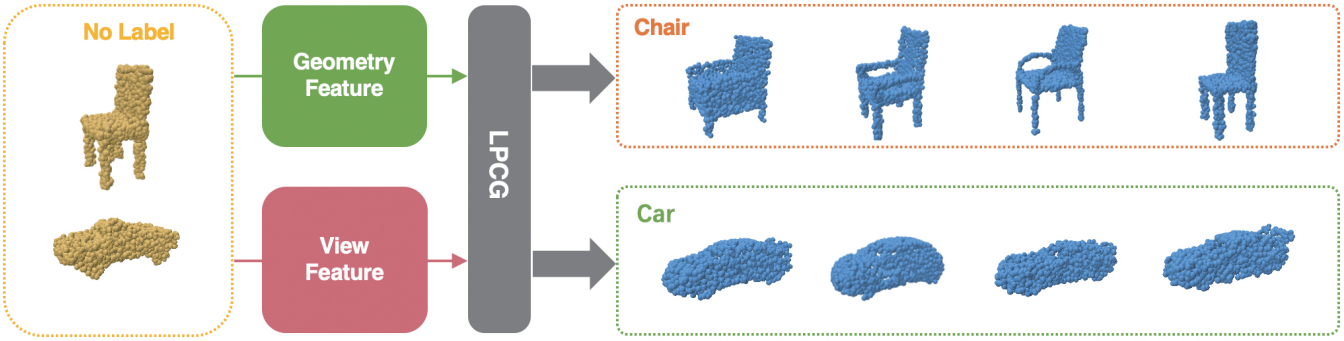

Figure 1: LPCG is a point cloud generation architecture that enabling training with a single model to generate high-quality 3D point clouds of various classes, as well as labeling them. Our model integrates the geometric feature and view feature of the point cloud to achieve the SOTA performance for point cloud generation tasks.

view feature with custom label pool to get label for the generated point cloud.

In this paper, we aim to make the following contributions:

- We propose LPCG, a self-conditional architecture that is capable of generating diverse and high-quality point clouds without any label input. Our method contains multimodal extraction module, feature diffusion modules and generation module.
- We design a multimodal fusion module that integrates 3D geometric features with 2D visual representation of point cloud view, thereby improving the generation quality and enabling the annotation of the generated point clouds.
- We demonstrate that LPCG achieves state-of-the-art performance on the ShapeNet dataset, surpassing existing methods and thereby setting a new benchmark for self-conditioned point cloud generation.

Related Work

3D Shape Generation

3D shape generation aims to synthesize high-fidelity 3D assets such as point clouds. The methods of 3D shape generation generally fall into two categories: conditional generation and unconditional generation. Conditional generation refers to generating content by utilizing specific conditions or contextual information to guide and constrain the generation process. For instance (Fan, Su, and Guibas 2017; Kurenkov et al. 2018; Gao et al. 2022), using 2D images as conditional inputs to generate corresponding 3D shapes has demonstrated the effectiveness of image-guided 3D shape generation. PVD (Zhou, Du, and Wu 2021) employs a point-voxel diffusion model to complete 3D shapes using partial point clouds and voxels as conditions which have shown strong 3D shape generation capabilities. LION (Vahdat et al. 2022) introduced a latent point diffusion model, which provides image or text-based conditional generation, showcasing diversity in generation. DiT-3D (Mo et al. 2024) employs labels as conditional inputs, delving into the utilization of DiT in 3D shape generation, thereby amplifying the quality and consistency of the generated shapes. In contrast, unconditional generation refers to models generating new samples without any external conditional input, based solely on their training data and learned patterns. Several methods (Achlioptas et al. 2018; Li et al. 2018; Shu, Park, and Kwon 2019; Wen, Yu, and Tao 2021) utilize Generative Adversarial Networks (GANs) to generate 3D point clouds, where adversarial training between the generator and discriminator enhances the quality of the generated point clouds. Alternatively, other methods employ normalizing flow to generate point clouds. PointFlow (Yang et al. 2019) generates 3D point clouds by learning the latent distribution of point clouds, while SoftFlow (Kim et al. 2020) performs transformations on manifolds for unsupervised learning. Regarding autoregressive models, PointGrow (Sun et al. 2020) generates high-quality 3D point clouds incrementally by employing autoregressive models and self-attention mechanisms, ensuring the coherence and consistency of the generated shapes. FoldingNet (Yang et al. 2018) employs an autoencoder framework to encode point clouds into latent representations, which are later decoded through deep grid deformation operations to produce high-quality 3D point clouds, achieving unconditional point cloud generation.

However, conditional generation algorithms are commonly impeded by the need of additional paired data, which is expensive to acquire. In terms of unconditional generation, particularly in multi-category generation tasks, prevailing methods face limitations as they only generate complete point clouds without categorizing the generated point clouds. Our approach aims to address these dual challenges by utilizing features extracted from raw data as conditional embeddings and harnessing large-scale 2D models for annotation. This approach eliminates the dependence on data labels and ensures high-quality generated data.

Self-Conditional Generation

Several recent work (Bao et al. 2022; Donahue and Simonyan 2019; Li et al. 2023; Lučić et al. 2019) showed that there is a significant gap between conditional and unconditional generation. To bridge this gap, self-conditional generation has emerged. In the field of 2D image generation, some methods (Bao et al. 2022; Hu et al. 2023; Liu et al. 2020) group images into clusters in the representa-

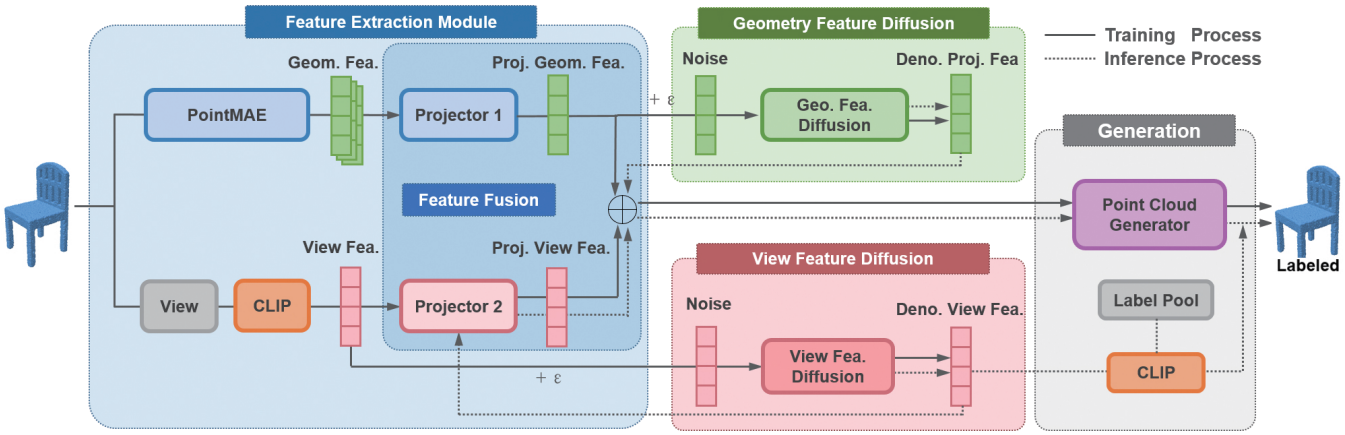

Figure 2: Overview of the LPCG. Firstly, the feature extraction modules extract multimodal features which are then fused and used as label substitutes for conditional point cloud generation. These fused features serve as conditions for the point cloud generator to produce point clouds. The feature diffusion modules provide diverse feature for guiding generation. Simultaneously, the sampled view features are inputted into CLIP to generate labels for the point clouds.

tion space and use these clusters as labels to guide generation. However, this often requires the number of clusters to be close to the number of classes. Other methods (Bordes, Balestriero, and Vincent 2021; Casanova et al. 2021) extract representations from existing images as conditional guidance. RCG (Li, Katabi, and He 2023) proposes a self-conditional generation method based on image representations, utilizing representation diffusion to generate new image representations. This method has achieved high performance in the field of 2D image generation. Inspired by this, LPCG applies the self-conditional generation into 3D shape creation task, and introduces a new architecture to generate diverse and high-quality labeled 3D point clouds.

Diffusion Models

In recent years, diffusion models (Ho, Jain, and Abbeel 2020; Song et al. 2020; Song, Meng, and Ermon 2020) have demonstrated exceptional performance in various generative tasks, such as 3D generation (Mo et al. 2024), image generation (Dhariwal and Nichol 2021), text generation (Hoogeboom et al. 2021; Austin et al. 2021), speech generation (Kong et al. 2020; Chen et al. 2020), and video generation (Ho et al. 2022; Harvey et al. 2022). Diffusion models generate new data by progressively adding noise to the data, and then learn how to reverse this process. During the forward diffusion process, the model gradually adds Gaussian noise to the original data to create noisy data. In the reverse generation process, the model progressively removes the noise to restore the original data. This process involves training a neural network to learn the data distribution, and then sample from the distribution as latent vector to reconstruct the data, e.g., image or point clouds. Instead of learning the data distribution, LCPG leverages diffusion process as a tool for learning the geometry and view feature distributions in the training process. Then in the inference stage, our model samples from the distribution to get fused features, as to guide high-quality point cloud 3D shapes generation, as well as generating the corresponding labels. As such, our architecture is able to improve the labels 3D shape generation quality and support feature interpolation for 3D shape generation.

Method

Architecture Overview

We aim to train a model with unlabeled point clouds and generate high-quality labeled 3D point cloud shapes. To achieve this goal, we design a generation architecture that contains the following three modules:

- **Feature Extractor Module**: to extract geometry and view multimodal features for the point clouds and then fuse them;
- **Feature Diffusion Modules**: include geometric feature diffusion module and view feature diffusion module. The goal of these modules is to provide diverse feature for guiding inference stage to generate high-quality labelled point clouds;
- **Generation Module**: to generate point clouds with a point cloud generator and annotate them with CLIP.

It should be noted that our architecture applies different processes for training and inference stage. For training process, feature extractor plays a role in extracting the 3D geometry feature and 2D view features, and then integrate them to generate a fused feature. The fused feature is then used to train 3D point cloud shape generation. At the same time, we leverage the geometric and single view feature to train diffusion models to generate features, which supports self-conditional point cloud generation in the inference process.

For the inference process, there are two sampling ways to utilize our model, i.e., *Dataset Sampling Strategy(DataS)* and *Diffusion Sampling Strategy(DiffS)*. For *Dataset Sampling Strategy*, LPCG facilitates the extraction of geometric and single view features from existing 3D point clouds,

which are then fed into the generation module to produce 3D point clouds. This process is same as the *Training Process* as shown in Figure 2. In this way, our model can effectively leverage existing 3D point clouds to customize the style of new 3D point clouds, as elaborated in the feature interpolation section. For *Diffusion Sampling Strategy*, LPCG first generates view and geometry features with trained feature diffusion models. Then the features are fused and fed into the 3D point clouds generator to get the fianl 3D shapes. This process is detailed in Figure 2 as the *Inference Process*. By this way, LPCG is able to create diverse 3D point clouds that may not exist in the original dataset.

Feature Extractor Module

The goal of the feature extractor module is to capture the 3D geometric features of the input point cloud along with its 2D rendering view features, and then fuse them together. We introduce them in detail as follows.

The 3D geometric feature is extracted with a pretrained point cloud model. Specifically, we utilize the Point-MAE (Pang et al. 2022) to train the point cloud model to predict the masked parts of the point cloud, effectively capturing the geometric structure information. The point cloud model follows the original reconstruction setting of Point-MAE. After the model training, we employ the pretrained point cloud model to extract geometric features for the given point cloud. We take the output from the decoder of the point cloud model as the geometric features $f_g \in R^{H \times C1}$, where H represents the token number and $C1$ represents the channel number of each token. We normalize each feature with its own mean and variance.

Apart from the 3D geometric feature, we also extract 2D view features by leveraging the CLIP's strong feature extraction and image classification zero-shot ability. In our proposed method, we tackle the challenging 3D generation problem without any point cloud labels and only 3D point cloud is available for training. To obtain the 2D view features of point cloud, we use Polyscope to render the point clouds from a fixed viewpoint at 900×1000 resolution, and then process it with the CLIP to obtain view features $f_v \in R^{1 \times C2}$.

Multimodal Fusion Module. To utilize the 3D geometric features and the 2D view features to guide generation, we propose a multimodal fusion module to fuse these features as a substitute for the label embedding condition in training the conditional point cloud generator.

We design two projectors to align feature dimensions and followed by a feature addition step to fuse these features. Specifically, the projectors are two MLP layers which process the geometric feature $f_g \in R^{H \times C1}$ and the view feature $f_v \in R^{1 \times C2}$, to a projected geometric feature $F_g \in R^{1 \times D}$ and projected view feature $F_v \in R^{1 \times D}$. In this way, we reduce the dimension of the geometric feature from a multi-dimensional vector to a one-dimensional vector, which simplifies subsequent modeling for feature diffusion. Additionally, we projected the geometric feature and the view feature to the same size to facilitate feature fusion, where D is the dimension of projected feature. Finally, we add the two projected features together to obtain the fused feature, which serves as the label embedding to guide the generation.

Feature Diffusion Modules

To consistently acquire diverse features as conditional inputs for the generative inference phase, we draw inspiration from RCG and employ feature diffusion for feature generation. In PLCG, we train two distinct feature diffusion modules to produce feature.

Geometry Feature Diffusion Module. As shown in Figure 2, we first train Geometic Feature Diffusion by sampling from the 3D geometric projected feature space. Specifically, we first add noise for the 3D geometric projected feature from PointMAE and *Projector 1*. Finally, we train the geometry feature diffusion by recovering the original geometric projected feature from noised added feature. After training is completed, the geometric diffusion is utilized to generated denoised feature for the following 3D generation. Note that we trained 3d geometric projected feature. Compared to 3D geometric features, 3D geometric projected features have a simpler structure and lower memory requirements, significantly reducing the difficulty and cost of training.

View Feature Diffusion Module. Meanwhile, we train view feature diffusion by sampling from the 2D view feature space. Firstly, we add noise for the 2D view feature from CLIP. Secondly, we train the view feature diffusion by recovering the original 2D view feature from noised added feature. Additionally, to ensure consistency between the generated 3D geometric projected features and the 2D view features, we condition the 2D view feature generation process on the 3D geometric projected features of the same point cloud. Note that we train 2D view feature unlike before. Compared to 2D view projected features, 2D view features contain the original view features extracted by CLIP, which can be used for subsequent annotation processing.

Generation Module

The generation module consists of point cloud generator and label generation. The point cloud generator adopts the DiT-3D because of the efficient and scalable transformer diffusion. The point cloud label is generated by applying CLIP to classify the view features. This paper exclusively employs DiT-3D as the generator, but substituting it with another generator is straightforward.

During the training, our goal is to train the point cloud generator. Specifically, we first add Gaussian noise to a given point cloud. Secondly, the point cloud generator is trained by removing the noise to recover the original point cloud, conditioned on the fused feature of the same point cloud. The label annotation utilize the CLIP without any training. During inference, the point cloud generator needs fused feature as condition to guide generation. LPCG provides two sampling methods to get fused feature. We introduce them in details.

Dataset Sampling Strategy (DataS). The dataset sampling method samples 3D geometric projected features and 2D view features from the dataset. Specifically, we use the multimodal feature extractor and multimodal fusion module mentioned above to extract the 3D geometric projected

Method	Chair				Airplane				Car			
	1-NNA (↓)		COV (↑)		1-NNA (↓)		COV (↑)		1-NNA (↓)		COV (↑)	
	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD
r-GAN	83.69	99.70	24.27	15.13	98.40	96.79	30.12	14.32	94.46	99.01	19.03	6.539
l-GAN (CD)	68.58	83.84	41.99	29.31	87.30	93.95	38.52	21.23	66.49	88.78	38.92	23.58
l-GAN (EMD)	71.90	64.65	38.07	44.86	89.49	76.91	38.27	38.52	71.16	66.19	37.78	45.17
PointFlow	62.84	60.09	48.38	40.41	70.54	58.10	46.88	50.00	58.10	56.25	46.88	50.00
SoftFlow	59.21	60.05	41.39	43.40	76.05	65.00	46.91	47.90	64.77	60.09	42.90	44.60
SetVAE	58.84	65.07	46.83	44.83	76.54	67.65	47.95	47.90	59.94	59.94	49.15	46.59
DPF-Net	62.00	58.35	44.71	39.09	78.62	62.35	44.17	48.89	62.35	54.48	45.74	49.43
DPM	60.05	74.77	44.86	35.70	76.42	68.91	48.64	38.41	68.89	79.97	44.03	34.94
PVD	57.09	68.07	36.68	49.42	73.82	67.08	41.39	52.29	54.55	53.83	41.19	50.56
LION	53.70	52.34	48.94	52.11	67.41	61.23	49.63	50.00	53.41	51.14	50.00	56.53
DiT-3D-S	57.79	57.14	49.25	51.98	68.85	61.71	47.83	50.51	61.06	56.38	43.75	52.60
DiT-3D-XL	**49.11**	**50.73**	52.45	54.32	62.35	58.67	53.16	**54.39**	**48.24**	49.35	**50.00**	**56.38**
LPCG(DiT-3D-S)-DataS	51.77	53.62	**56.39**	**58.38**	**61.43**	**53.47**	**55.94**	51.66	51.69	**48.82**	**50.00**	54.16
LPCG(DiT-3D-S)-DiffS	54.68	54.38	48.59	52.25	65.09	57.92	54.59	53.45	58.72	52.60	44.53	48.95

Table 1: Comparison results (%) on shape metrics of LPCG and baseline models.

features $F_{gtrain} \in R^{N\times D}$ and 2D view features $F_{vtrain} \in R^{N\times C1}$ where N represents the number of samples in the dataset. During the inference stage, we randomly sample geometric projected features $F_{gtest} \in R^{M\times D}$ and view features $F_{vtest} \in R^{M\times C1}$, where M represents the number of point cloud to be generated. Due to the probabilistic characteristic of the generative models, our dataset sampling method can generate highly diverse data samples.

Diffusion Sampling Strategy (DiffS). The diffusion sampling method samples 3D geometric and 2D view features by using the above trained feature diffusion modules. Because the sampling features are diverse, we can obtain diverse conditions for the point cloud generator and obtain diverse generated point clouds. Specifically, we use the previously trained feature diffusion model to generate new 3D geometric projected features from noise, which are then used as conditional inputs to guide the generation of 2D view features. In this way, we can continuously obtain consistent and diverse features. Finally, the geometric projected features and view features are fed into a multimodal fusion module to generate fused features that guide the generation process. It is important to note that the geometric projected features no longer require processing by a projected layer.

Label Generation. Notably, in LPCG, CLIP is used to extract view features of the point cloud, which is used for training the sampling diffusion described above. Therefore, the view features obtained with these sampling methods can be subsequently recognized by CLIP. As shown in Figure 2, we provide a custom label pool which includes pre-defined categories. The sampled view features are not only used to guide generation but also fed into CLIP for matching corresponding categories through similarity calculation, thereby assigning labels to the point clouds generated based on these features.

Experiment

Experimental Setup

Datasets. Following previous methods (Yang et al. 2019; Mo et al. 2024; Vahdat et al. 2022; Zhou, Du, and Wu 2021), we used ShapeNet (Chang et al. 2015) as our dataset. During the feature extractor pre-training phase, we referred to the processing procedure of point-MAE and sampled 1024 points for each point cloud of ShapeNet. In the training phase of the point cloud generator, we referred to the data processing procedure of DiT-3D and sampled 2048 points for each point cloud of ShapeNet. Following PointFlow (Yang et al. 2019), we also perform global normalization.

Evaluation Metrics. To perform comprehensive comparisons, we adopted the evaluation metrics commonly used in previous works. Specifically, we used Chamfer Distance (CD) and Earth Mover's Distance (EMD) as the primary distance metrics to calculate 1-Nearest Neighbor Accuracy (1-NNA) and Coverage (COV). These metrics are crucial for assessing the quality of generative models. 1-NNA evaluates the leave-one-out accuracy of the 1-NN classifier, reflecting the quality and diversity of the generated point clouds. A lower 1-NNA indicates better performance. COV measures the number of reference point clouds that match at least one generated shape, indicating the diversity of the generated point clouds. Generally, a higher COV is better, showing greater diversity, but it does not necessarily indicate higher quality.

Baselines. We compare with two kinds of point cloud generation approaches: conditional generation and unconditional generation. The approaches are listed as follows. 1) r-GAN (Achlioptas et al. 2018), the model based on GAN; 2) PointFlow (Yang et al. 2019), which achieves high-quality 3D point cloud generation by incorporating continuous normalizing flows (CNF) and a dual network architecture. 3) SoftFlow (Kim et al. 2020), the model incorporates continu-

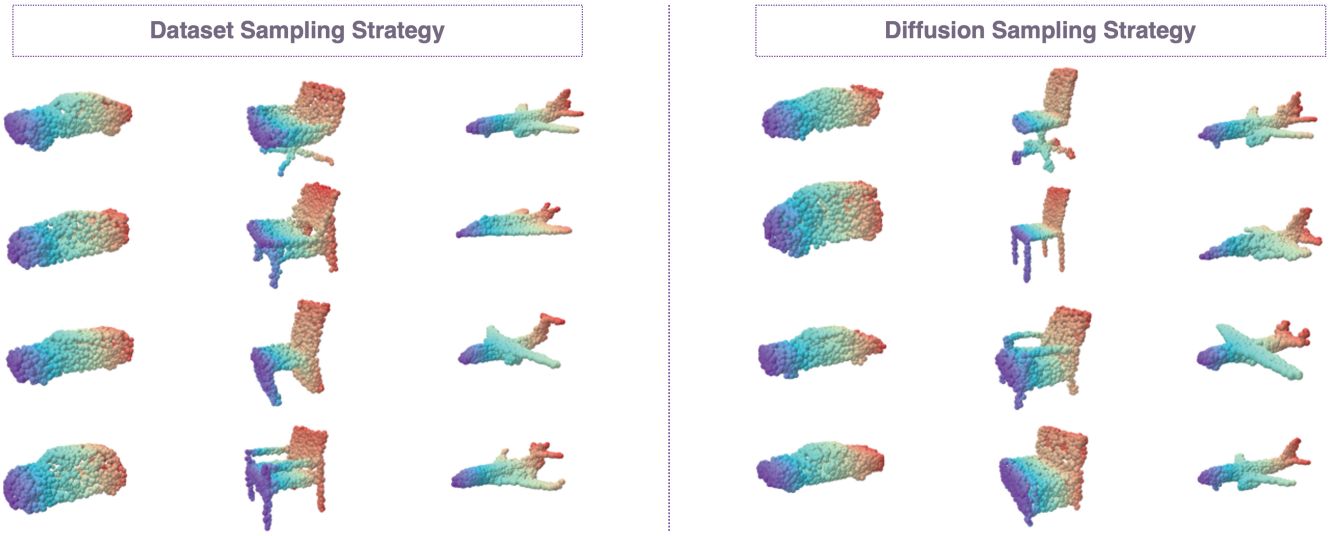

Figure 3: Examples of point clouds generated by LPCG. The left three columns are generated by *Dataset Sampling Strategy*, and the right three columns are generated by *Diffusion Sampling Strategy*.

ous normalizing flows and a novel soft deformation mechanism. 4) SetVAE (Kim et al. 2021), the model incorporates a variational autoencoder (VAE) and a permutation-invariant set representation. 5) DPF-Net (Klokov, Boyer, and Verbeek 2020), which achieves 3D point cloud generation by incorporating a dual-path framework and an attention mechanism. 6) DPM (Ho, Jain, and Abbeel 2020), the model uses a deep generative way. 7) PVD (Zhou, Du, and Wu 2021), based on Point-Voxel diffusion. 8) LION (Vahdat et al. 2022), which uses latent sapce diffusion. 9) DiT-3D (Mo et al. 2024), an effective DiT model for 3D object generation.

Experimental Result

Quantitative Results. As shown in Table 1, experiments show that LPCG has promising point cloud generation performance. Specifically, when using *Dataset Sampling Strategy* and based on DiT-3D-S, we achieved better performance across all metrics than previous works, except for DiT-3D-XL and LION. Our architecture outperforms 10 out of 12 metrics than LION. Compared to the original Dit-3D-XL which has 20 times more parameters than DiT-3D-S, we have achieved better results on 7 out of 12 metrics, with merely increasing 6.1MB parameters. This indicates that LPCG can achieve better generation results with a smaller model size by introducing multimodal feature representative labels through self-conditioning is an effective method for generating high-quality point clouds.

When using *Diffusion Sampling Strategy* and also based on DiT-3D-S, we can see from the table 1 that our method still demonstrates competitiveness. Overall, LPCG has 10 out of 12 metrics better than the original DiT-3D-S model.

Subsequently, we applied our method to multi-category unconditional generation, starting with training on two categories: chairs and cars. During the generative inference stage, we sample features using the two sample methods mentioned above to guide the generation of point clouds. Then we use CLIP to process 2D view feature to label the generated point clouds. As shown in Tables 2 and 3, we first specify the features of a single category for sampling, which guides the generation process. In both *Dataset Sampling Strategy* and *Diffusion Sampling Strategy*, the chair category achieved a 100% labeling accuracy, and advanced performance was obtained in terms of generation quality. However, when generating the point clouds of car category, the labeling accuracy decreased slightly. This decline is attributed to some chair feature being misidentified as belonging to the car category by CLIP during the inference. As a result, when sampling the 3D geometric projected features and 2D view features of the car class, some chair features were incorrectly sampled. Despite this, the generation qual-

Category	Label Accuracy	1-NNA (↓)		COV (↑)	
		CD	EMD	CD	EMD
Chair	100	53.91	55.22	51.67	51.85
Car	88	52.08	49.34	52.86	55.98
All	95.8	53.39	52.03	52.30	53.66

Table 2: The test results on shape metric and label accuracy of *Dataset Sampling Strategy* in multi-category training.

Category	Label Accuracy	1-NNA (↓)		COV (↑)	
		CD	EMD	CD	EMD
Chair	100	55.02	54.31	49.72	53.40
Car	92.3	61.84	55.46	48.69	51.82
All	97.44	54.27	52.58	49.80	54.91

Table 3: The test results on shape metric and label accuracy of *Diffusion Sampling Strategy* in multi-category training.

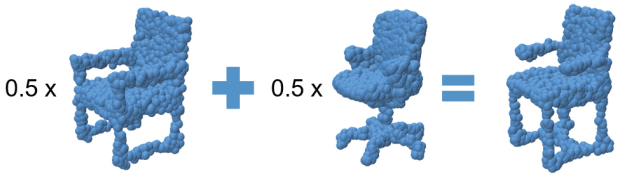

Figure 4: Visual results of interpolated features. With feature based methods, LPCG is capable of fusion features of different 3D point cloud shapes.

ity remained high. In the multi-category guided generation, the overall labeling accuracy was high, and the generation performance was excellent.

Feature Interpolation. Our LPCG is conditioned on 3D geometric features and 2D view features. To verify the spatial continuity of these 3D geometric features and 2D view features, we performed feature transfer by linearly interpolating the features of two point clouds. We applied a 50% weight linear interpolation to the given two point clouds, thus getting the interpolated point cloud. As shown in Figure 4, the interpolated point cloud still maintains a realistic appearance. The point cloud on the right smoothly combines the shape structures of both point clouds, retaining the overall shape of point cloud 1 while integrating the handle and chair legs from point cloud 2. This demonstrates that the introduction of interpolation can help us generate more diverse point clouds.

Ablations

In this section, we conducted ablation studies to demonstrate the effectiveness of the introduced feature replacement. Additionally, we performed extensive experiments to explore the impact of the point cloud generator and feature diffusion. Here, we used DiT-3D/S to train the car category as the default setting.

The Impact of Multimodal Features. To verify the effectiveness of multimodal fusion features in guiding generation, we conducted an ablation study on self-conditioned features. As shown in the table 4, the overall generation performance is best when both types of features are included. The performance significantly declines when either geometric or view features are missing, and it is the worst when both features are removed. Additionally, without view features, we are unable to label the point clouds. It explains that the method we introduced for training the model using multimodal features for labels is effective.

Conditional Features	1-NNA (↓)		COV (↑)	
	CD	EMD	CD	EMD
Geometric and View	50.52	48.56	54.16	56.55
Geometric	53.48	47.68	50.06	55.07
View	54.81	53.12	52.34	55.20
-	59.50	55.08	43.75	50.26

Table 4: The test results on shape metric of *Diffusion Sampling Strategy* under different conditional features.

Point Cloud Generator	1-NNA (↓)		COV (↑)	
	CD	EMD	CD	EMD
DiT-3D-S	50.91	52.34	48.95	50.78
DiT-3D-L	49.60	51.82	53.12	52.52

Table 5: The test results for different point cloud generators.

Training Epoch	1-NNA (↓)		COV (↑)	
	CD	EMD	CD	EMD
50	56.77	54.55	44.01	51.56
100	57.29	51.17	45.31	51.56
300	52.08	52.27	49.74	52.08
500	48.56	47.26	49.73	53.12

Table 6: The test result on shape metric for different epoch.

Applicable to Different Point cloud Generators. Point cloud generator is important in LPCG. To evaluate the impact of different point cloud generator on LPCG, we conduct training on DiT-3D-S and DiT-3D-L under the same settings. As shown in the table 5, LPCG achieved better generation quality when using DiT-3D-L. This indicates that LPCG is applicable to different point cloud generators, and strong point cloud generators can get better generation quality.

The Impact of Feature Diffusion. We further explored the impact of feature diffusion on the generative results. We evaluated the final generated results by adjusting the training epochs of feature diffusion. As shown in table 6, our generation quality improves when the number of training epochs increases.

Conclusion

In this paper, we design a self-conditioned point cloud generation algorithm. Our method uses geometric features and view features as labels for training, avoiding the cost of manual data labeling, and achieves high-quality generation results. Additionally, our view features are extracted by the 2D large model CLIP, which can compute with a predefined text library to obtain the labels corresponding to the point clouds generated based on these features. Experiments show that our method achieves better generation results than those methods trained directly with labels. Meanwhile, our method achieves an annotation accuracy of 97% for the generated results, thanks to the powerful capabilities of 2D large model CLIP.

Acknowledgments

This research work was supported by the Fundamental Research Funds for the Central Universities and the Natural Science Basic Research Program of Shaanxi Province under Grant 2024JC-YBQN-0702.

References

Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; and Guibas, L. 2018. Learning representations and generative models for

3d point clouds. In *International conference on machine learning*, 40–49. PMLR.

Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and Van Den Berg, R. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34: 17981–17993.

Bao, F.; Li, C.; Sun, J.; and Zhu, J. 2022. Why are conditional generative models better than unconditional ones? *arXiv preprint arXiv:2212.00362*.

Bordes, F.; Balestriero, R.; and Vincent, P. 2021. High fidelity visualization of what your self-supervised representation knows about. *arXiv preprint arXiv:2112.09164*.

Casanova, A.; Careil, M.; Verbeek, J.; Drozdzal, M.; and Romero Soriano, A. 2021. Instance-conditioned gan. *Advances in Neural Information Processing Systems*, 34: 27517–27529.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chen, N.; Zhang, Y.; Zen, H.; Weiss, R. J.; Norouzi, M.; and Chan, W. 2020. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

Donahue, J.; and Simonyan, K. 2019. Large scale adversarial representation learning. *Advances in neural information processing systems*, 32.

Fan, H.; Su, H.; and Guibas, L. J. 2017. 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*, 605–613.

Gao, J.; Shen, T.; Wang, Z.; Chen, W.; Yin, K.; Li, D.; Litany, O.; Gojcic, Z.; and Fidler, S. 2022. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35: 31841–31854.

Harvey, W.; Naderiparizi, S.; Masrani, V.; Weilbach, C.; and Wood, F. 2022. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35: 27953–27965.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022. Video diffusion models. *Advances in Neural Information Processing Systems*, 35: 8633–8646.

Hoogeboom, E.; Gritsenko, A. A.; Bastings, J.; Poole, B.; Berg, R. v. d.; and Salimans, T. 2021. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*.

Hu, V. T.; Zhang, D. W.; Asano, Y. M.; Burghouts, G. J.; and Snoek, C. G. 2023. Self-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18413–18422.

Kim, H.; Lee, H.; Kang, W. H.; Lee, J. Y.; and Kim, N. S. 2020. Softflow: Probabilistic framework for normalizing flow on manifolds. *Advances in Neural Information Processing Systems*, 33: 16388–16397.

Kim, J.; Yoo, J.; Lee, J.; and Hong, S. 2021. SetVAE: Learning Hierarchical Composition for Generative Modeling of Set-Structured Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15059–15068.

Klokov, R.; Boyer, E.; and Verbeek, J. 2020. Discrete point flow networks for efficient point cloud generation. In *European Conference on Computer Vision*, 694–710. Springer.

Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; and Catanzaro, B. 2020. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*.

Kurenkov, A.; Ji, J.; Garg, A.; Mehta, V.; Gwak, J.; Choy, C.; and Savarese, S. 2018. Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 858–866. IEEE.

Li, C.-L.; Zaheer, M.; Zhang, Y.; Poczos, B.; and Salakhutdinov, R. 2018. Point cloud gan. *arXiv preprint arXiv:1810.05795*.

Li, T.; Chang, H.; Mishra, S.; Zhang, H.; Katabi, D.; and Krishnan, D. 2023. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2142–2152.

Li, T.; Katabi, D.; and He, K. 2023. Self-conditioned image generation via generating representations. *arXiv preprint arXiv:2312.03701*.

Liu, S.; Wang, T.; Bau, D.; Zhu, J.-Y.; and Torralba, A. 2020. Diverse image generation via self-conditioned gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14286–14295.

Lučić, M.; Tschannen, M.; Ritter, M.; Zhai, X.; Bachem, O.; and Gelly, S. 2019. High-fidelity image generation with fewer labels. In *International conference on machine learning*, 4183–4192. PMLR.

Mo, S.; Xie, E.; Chu, R.; Hong, L.; Niessner, M.; and Li, Z. 2024. Dit-3d: Exploring plain diffusion transformers for 3d shape generation. *Advances in Neural Information Processing Systems*, 36.

Pang, Y.; Wang, W.; Tay, F. E.; Liu, W.; Tian, Y.; and Yuan, L. 2022. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, 604–621. Springer.

Shu, D. W.; Park, S. W.; and Kwon, J. 2019. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3859–3868.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Sun, Y.; Wang, Y.; Liu, Z.; Siegel, J.; and Sarma, S. 2020. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 61–70.

Vahdat, A.; Williams, F.; Gojcic, Z.; Litany, O.; Fidler, S.; Kreis, K.; et al. 2022. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35: 10021–10039.

Wen, C.; Yu, B.; and Tao, D. 2021. Learning progressive point embeddings for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10266–10275.

Wu, S.; Lin, Y.; Zhang, F.; Zeng, Y.; Xu, J.; Torr, P.; Cao, X.; and Yao, Y. 2024. Direct3D: Scalable Image-to-3D Generation via 3D Latent Diffusion Transformer. *arXiv preprint arXiv:2405.14832*.

Yang, G.; Huang, X.; Hao, Z.; Liu, M.-Y.; Belongie, S.; and Hariharan, B. 2019. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4541–4550.

Yang, Y.; Feng, C.; Shen, Y.; and Tian, D. 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 206–215.

Zhou, C.; Zhong, F.; Hanji, P.; Guo, Z.; Fogarty, K.; Sztrajman, A.; Gao, H.; and Oztireli, C. 2023. FrePolad: Frequency-Rectified Point Latent Diffusion for Point Cloud Generation. *arXiv preprint arXiv:2311.12090*.

Zhou, L.; Du, Y.; and Wu, J. 2021. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*, 5826–5835.