SP-GAN: Sphere-Guided 3D Shape Generation and Manipulation[1]

RUIHUI LI,XIANZHI LI,KA-HEI HUI,and CHI-WING,The Chinese University of Hong Kong,China

Abstract

SP-GAN is a new unsupervised sphere-guided generative model for direct synthesis of 3D shapes in the form of point clouds presented by this paper. SP-GAN is able to synthesize diverse and high-quality shapes with fine details in the form of point clouds. In advance, SP-GAN promote controllability for part-aware shape manipulation, while trainable without any parts annotations.SP-GAN combine a global prior which is to guide the generation process and a local prior which is to provide details.The key insight in the paper is to decompose the entire 3D generation task into two parts: the first part is modeling the overall shape, and the second part is adjusting local structure, which helps simplify the learning process and enhance the quality of generative shapes. Moreover, SP-GAN enable to establish an implicit dense correspondence between the sphere points and points in every generative shapes, which allows various forms of structure-aware shape manipulations such as part editing, part-wise shape interpolation, multi-shape part composition and part co-segmentation.

Keywords:shape analysis and synthesis, generative model,3D shape generation, 3D shape generation, 3D shape manipulation, point clouds

1 Introduction

A challenging problem in 3D shape creation is how to build generative models to synthesize new, diverse, and realistic-looking 3D shapes, while having structure-aware generation and manipulation. This paper design a deep generative models to characterize the shape distribution and learn to directly generate shapes in an unsupervised manner. This paper presents a new GAN model called SP-GAN,in which the authors use a Sphere as Prior to guide the direct generative process of 3D shapes(in the form of point cloud).Their approach enable to generate point clouds with finer details and less noise compared with the state-of-the-art. Importantly, the design facilitates controllability in the generative process, since the model implicitly embeds a dense correspondence between the generated shapes, and training this model is unsupervised, without requiring any parts annotations. Based on this, generation and manipulation of shapes with part awareness can be performed. The key design in SP-GAN is to design the generator with two decoupled ibputs:(i) a global prior S,which is a fixed unit sphere in the form of point cloud, to process;and(ii) a local prior z, which is a random latent code to provide local details, rather than having a single latent code as input like the conventional generative models.These two inputs are packed into a prior latent matrix by attaching a latent code z to every point in S. The generative process starts from a shared global initialization,yet accommodating the spatial variations, such that every point moves towards its desired location for forming the shape. A key insight behind this design is that authors formulate the generation task as a transformation and disentangle the complex 3D shape generation task into (i)a global shape modeling and (ii)a local structure adjustment. Another important consequence is that this model facilitates structure controllability in the generative process, through an implicit dense correspondence between the input sphere and the generated shapes. The sphere provides a common working space for shape generation and manipulation. Modifing the latent vectors associated with specific points on S while keeping others unchanged, can manipulate local structures for the associated parts in the sphere. Moreover, since parts are connected geometrically and semantically with one another, changing one part may cause slight changes in some other parts for structural compatibility. Furthermore, with the dense correspondence, the latent code from different generated shapes can be interpolated to morph between shapes in a shape-wise or part-wise fashion.

2 Related works

SP-GAN model is unconditional which is inspired by some works using deep neural networks such as PointNet[2], DGCNN[3]. These methods can be roughly classified into autoregressive-based, flow-based, and GAN-based.

2.1 Autoregressive-based generative approach

Autoregressive-based generative approach such as PointGrow[4] models the joint 3D spatial distribution of points in a point cloud. Due to the iterative property intrinsic to autoregressive models, the model cannot scale well with the size of the point cloud.

2.2 Flow-based generative approach

Flow-based generative approach such as PointFlow[5],discrete point flow[6], SoftFlow[7], ShapeGF[8] learns to model the distribution of points in a shape mainly by an invertible parameterized transformation of the points. While substantial progress has been made, the invertibility constraint in flow-based approach unavoidably limits the representation capability of the models. Also, the learned parameterized transformation can be regarded as a rough estimation of the averaged training data distribution, so generated point samples tend to be blurry and noisy.

2.3 GAN-based generative approach

GAN-based generative approach explores adversarial learning to train the shape generation model with the help of a discriminator. Achlioptas et al.[9] introduce the first set of deep generative models to produce point clouds from a Gaussian noise vector, including an r-GAN that operates on a raw point cloud input and an l-GAN that operates on the bottleneck latent variables of a pretrained autoencoder. To overcome the redundancy and structural irregularity of point samples, Spectral-GANs[10] synthesize shapes using a spherical-harmonicsbased representation.tree-GAN[11] performs graph convolutions in a tree. PDGN[12] is a progressive deconvolution network to generate 3D point clouds.

3 Method

3.1 Overview

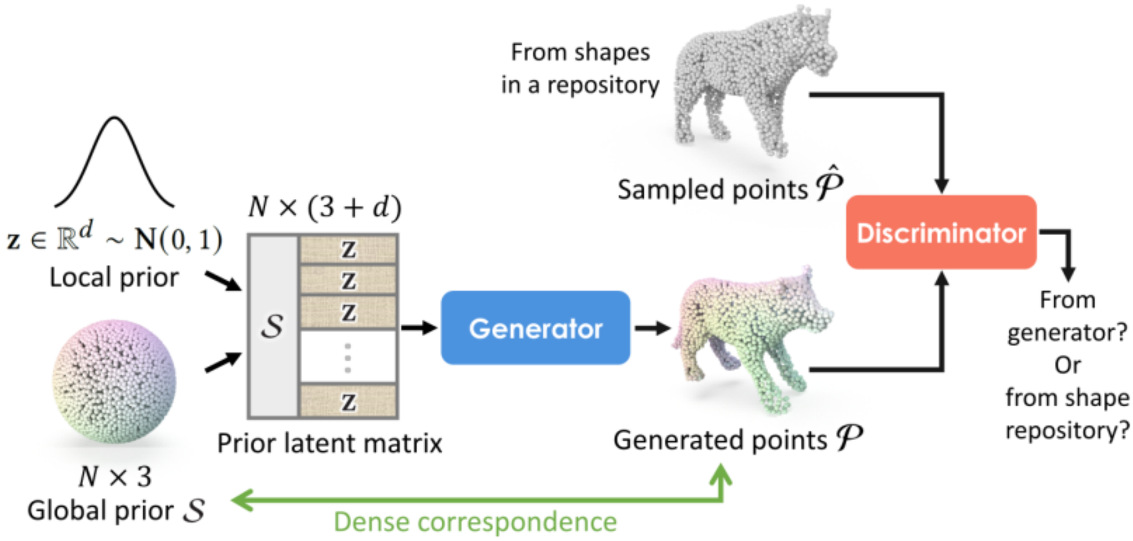

Figure 1: Overview of the method

The basic model of SP-GAN is illustrated in Figure 1,with shapes represented as point clouds. The generator consumes two decoupled inputs: a global prior S, which contains the 3D coordinates of N points uniformly on a unit sphere, and a local prior z, which is a d-dimensional random latent vector sampled from a standard normal distribution. One latent vector z is packed with each point in S to form the prior latent matrix as the generator input.

During training, the generator synthesize point cloud $P \in R^{N\times3}$ from S and z and sample another point cloud $\hat{P} \in R^{N\times3}$ from shape in a 3D repository. The discriminator should learn to differentiate P and $\hat{P}$. During testing, SP-GAN randomly samples a latent code and pack it with sphere S into a prior latent matrix, and feed it to the trained generator to produce a new shape.

3.2 Generator

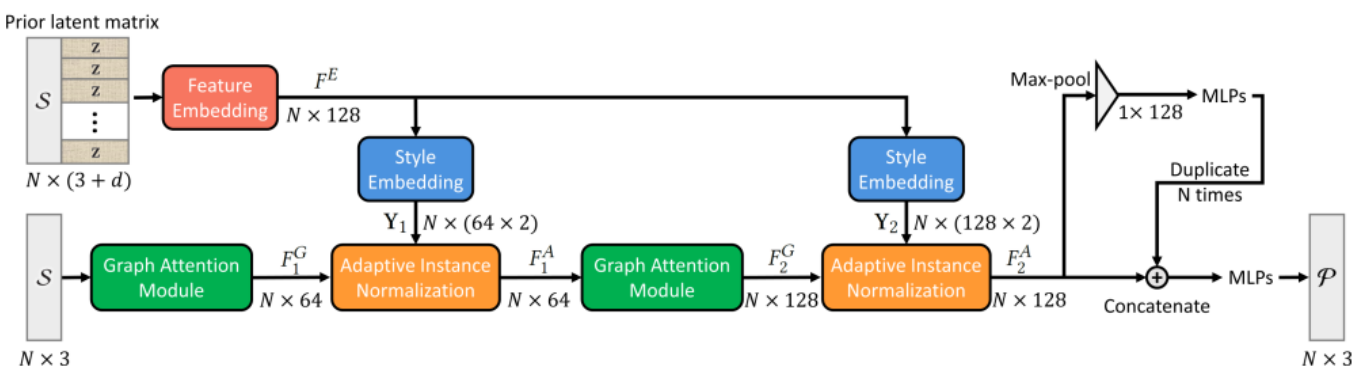

Figure 2: Architecture of the generator

The generator takes the prior latent matrix as input and feeds sphere S to the Graph Attention Module(green box) to extract the point-wise feature map F_1^G. On the top branch, a Feature Embeding(orange box) is used to extract local style F^E from the prior latent matrix. Style Embedding(blue box) embed the local style $F^E inY_1$. Then, to embed the local style in F_1^G, an Adaptive Instance Normalization is intruduced to produce F_1^A with richer local details. The generator repeat this process with another round of style embedding and normalization, and then follow PointNet[2] to reconstruct point cloud P from the embedded feature map F_2^A.

3.3 Discriminator

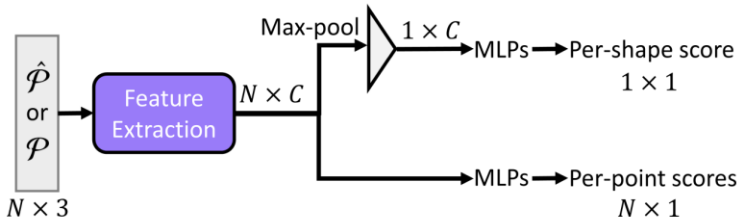

Figure 3: Architecture of the discriminator

The input of discriminator is either P produced by the generator or a sampled from a 3D repository. The discriminator would learn a $1 \times C$ global feature vector for the whole shape and predict a single score that indicates the source of the point cloud. Specifically, Feature Extraction(purple box) which adopt PointNet[2] as the backbone extract $N \times C$ point features. Then, the discriminator predict a per-shape score and per-point scores.

3.4 Loss

Two losses based on the least squares loss[13] are used:

$$\mathrm{L}_G = \frac{1}{2}[D(P) - 1]^2 \text{ (1)}$$

$$and$$

$$\mathrm{L}_G = \frac{1}{2}[(D(P) - 0)^2 + (D(\hat{P}) - 1)^2] \text{ (2)}$$

Hence, we model the discriminator loss L_D as a summation of shape loss L_D^{shape} and point loss L_D^{point}:

$$\mathrm{L}_D = L_D^{shape} + L_D^{point} \text{ (3)}$$

$$\mathrm{L}_D^{shape} = \frac{1}{2}[(D(P) - 0)^2 + (D(\hat{P}) - 1)^2] \text{ (4)}$$

$$\mathrm{L}_D^{point} = \frac{1}{2N}\sum_{i=1}^{N}[(D(p_i)-0)^2+(D(\hat{p}_i)-1)^2] \ (5)$$

is a balance parameter; p_i and $\hat{p}_i$ are the i-th point in P and $\hat{P}$.

Correspondingly, the objective for training the generator becomes

$$\mathrm{L}_G = \frac{1}{2}[D(P)-1]^2+\frac{1}{2N}\sum_{i=1}^{N}[D(p_i-1)]^2 \ (6)$$

4 Implementation details

4.1 Comparing with released source codes

I used a variety of data sets, so in order to unify the interface to use these data sets, I deleted the original data set processing code and wrote the data processing part of the algorithm.The algorithm is described as follows:

data process

Procedure 1 Data Set Processing

Input: Meshes *M or* Point Clouds *P*

Output: randomly sampled Point Clouds *P'*

for *m* ***in*** *meshes M or reconstructed meshes M' from Point Clouds P* **do**

| $p' = \mathrm{SAMPLE}(m)$ *or* $p' = \mathrm{SAMPLE}(m')$

end

$len = \mathrm{LEN(P')}$

for *epoch* ***in*** *epoches* **do**

| $index = randomly\ select\ 32\ numbers\ in\ [0, len-1]\$$

| $\mathrm{data}_{epoch} = P'[index]$

| $training\ steps$ ...

end

4.2 Experimental environment setup

I reoccurrence SP-GAN using PyTorch and train it on two NVidia TITAN Xp GPUs.

4.3 Interface design

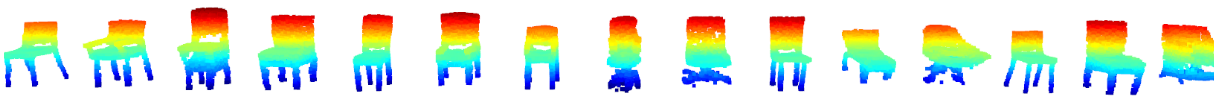

Figure 4: chairs

4.4 Main contributions

5 Results and analysis

There are some categories of point clouds generated by my trained generator.

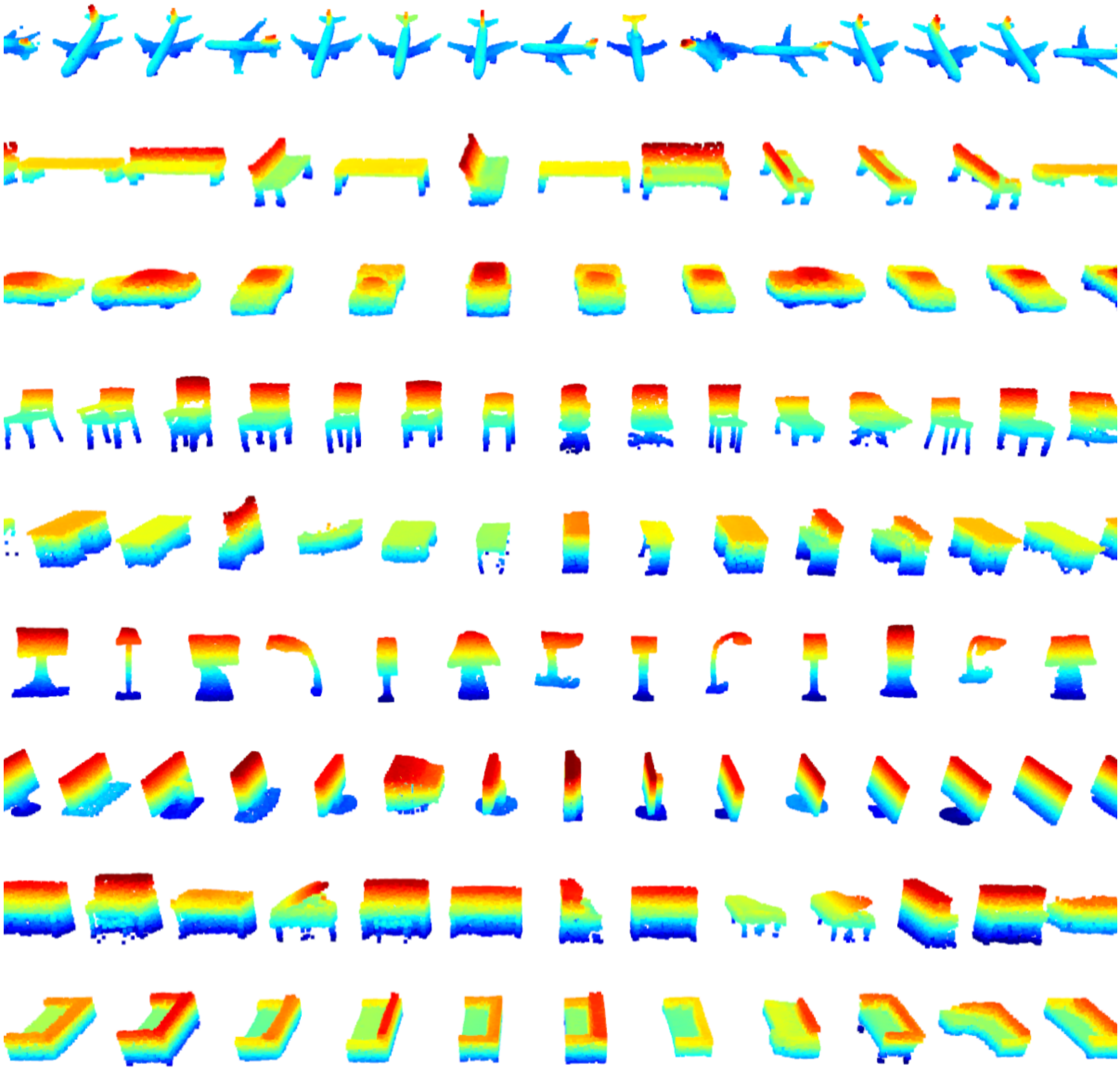

Figure 5: Experimental results

By interpolating the latent code of two point clouds of the same type, point clouds with their own characteristics can be generated at the same time. By modifying the weight of latent code of these two point clouds, the transition from one point cloud to another point cloud can be made.

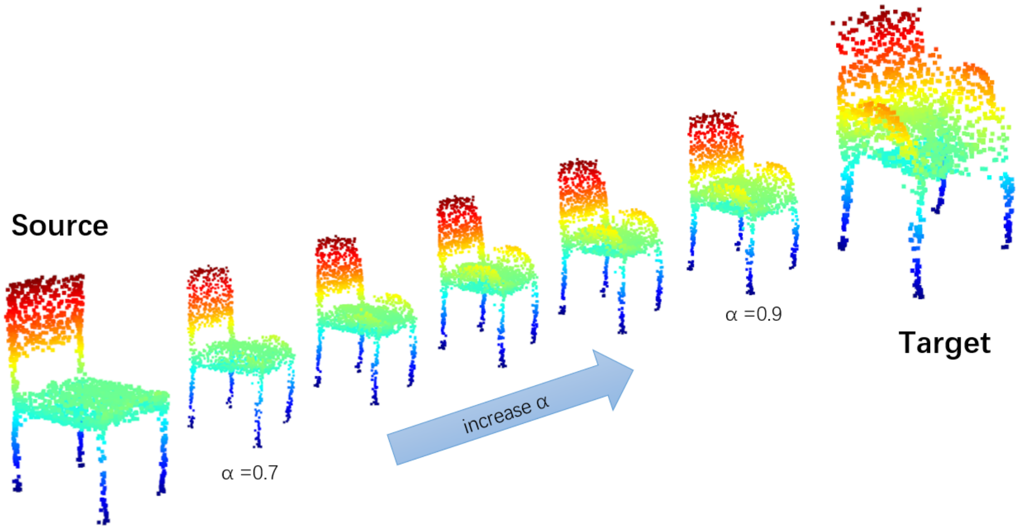

Figure 6: latent code interpolation

6 Conclusion and future work

SP-GAN decouples the input into a global prior(unit sphere) and a local prior(latent code), and formulate the generative process with style embedding and adaptive instance normalization to bring local style from latent code into the point features of the sphere points.Importantly, SP-GAN introduces structure controllability into the generative process through the implicit dense correspondence. So, we can modify or interpolate latent codes in a shape-wise or part-wise manner, and enable various forms of structure-aware shape manipulations that cannot be achieved by previous works on direct point cloud generation.However, for shapes with limited training samples or with complex or thin structures, the generated shapes may still be blurry (or noisy). In the future, we will focus on generating high quality point clouds with complex or thin structures.

References

[1] LI R, LI X, HUI K H, et al. SP-GAN: Sphere-guided 3D shape generation and manipulation[J]. ACM Transactions on Graphics (TOG), 2021, 40(4): 1-12.

[2] QI C R, SU H, MO K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.

[3] WANG Y, SUN Y, LIU Z, et al. Dynamic graph cnn for learning on point clouds[J]. Acm Transactions On Graphics (tog), 2019, 38(5): 1-12.

[4] SUN Y, WANG Y, LIU Z, et al. Pointgrow: Autoregressively learned point cloud generation with self-attention[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2020: 61-70.

[5] YANG G, HUANG X, HAO Z, et al. Pointflow: 3d point cloud generation with continuous normalizing flows[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 4541-4550.

[6] KLOKOV R, BOYER E, VERBEEK J. Discrete point flow networks for efficient point cloud generation

[C]//European Conference on Computer Vision. 2020: 694-710.

[7] KIM H, LEE H, KANG W H, et al. Softflow: Probabilistic framework for normalizing flow on manifolds [J]. Advances in Neural Information Processing Systems, 2020, 33: 16388-16397.

[8] CAI R, YANG G, AVERBUCH-ELOR H, et al. Learning gradient fields for shape generation[C]// European Conference on Computer Vision. 2020: 364-381.

[9] ACHLIOPTAS P, DIAMANTI O, MITLIAGKAS I, et al. Learning representations and generative models for 3d point clouds[C]//International conference on machine learning. 2018: 40-49.

[10] RAMASINGHE S, KHAN S, BARNES N, et al. Spectral-gans for high-resolution 3d point-cloud generation[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2020: 8169-8176.

[11] SHU D W, PARK S W, KWON J. 3d point cloud generative adversarial network based on tree structured graph convolutions[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 3859-3868.

[12] HUI L, XU R, XIE J, et al. Progressive point cloud deconvolution generation network[C]//European Conference on Computer Vision. 2020: 397-413.

[13] MAO X, LI Q, XIE H, et al. Least squares generative adversarial networks[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2794-2802.