Computational Visual Media
https://doi.org/10.1007/s41095-0xx-xxxx-x

Review Article

A Survey of Deep Learning-based 3D Shape Generation

Qun-Ce Xu[1], Tai-Jiang Mu[1](✉), and Yong-Liang Yang[2]

Abstract Deep learning has been successfully used for tasks in the 2D image domain. Research on 3D computer vision and deep geometry learning has also attracted attention. Considerable achievements have been made regarding feature extraction and discrimination of 3D shapes. Following recent advances in deep generative models such as generative adversarial networks, effective generation of 3D shapes has become an active research topic. Unlike 2D images with a regular grid structure, 3D shapes have various representations, such as voxels, point clouds, meshes, and implicit functions. For deep learning of 3D shapes, shape representation has to be taken into account as there is no unified representation that can cover all tasks well. Factors such as the representativeness of geometry and topology often largely affect the quality of the generated 3D shapes. In this survey, we comprehensively review works on deep-learning-based 3D shape generation by classifying and discussing them in terms of the underlying shape representation and the architecture of the shape generator. The advantages and disadvantages of each class are further analyzed. We also consider the 3D shape datasets commonly used for shape generation. Finally, we present several potential research directions that hopefully can inspire future works on this topic.

Keywords 3D representations, geometry learning, generative models, deep learning

1 Introduction

With the rapid development of 3D acquisition and modeling techniques [1], 3D data can be efficiently captured from the real world or created with easy-to-use modeling software. Furthermore, recent advances in Internet tools, especially online repositories, allow 3D shapes to be shared among users [2]. As a result, 3D shapes are largely available nowadays and have been widely used for important applications such as entertainment and games, robotics and autonomous systems, virtual and augmented reality.

Concurrently, with the explosive growth of data [2, 3] and the significantly enhanced power of modern computational devices [4], deep learning based on large-scale neural networks has become an active research area, and a fundamental technique in computer vision and computer graphics [5, 6]. Following the excellent results of applying deep learning to 2D images [7–9], researchers have also adopted deep learning for 3D shape understanding and processing. Many works based on deep geometry learning have demonstrate its superiority over traditional methods [10–12]. Since deep learning often requires a large amount of data to train models, building and benchmarking 3D shape datasets has also become particularly important [2, 13].

This survey aims to review the use of deep learning techniques to generate 3D geometric shapes (see Table 1). Compared to traditional 3D acquisition and modeling techniques that focus on the precision and quality of the resultant shapes via hand-crafted features and algorithms, deep-learning-based 3D shape generation has the advantage of learning a complicated yet comprehensive latent space of 3D shapes. Thus, it enables more creative generation and exploration of novel shapes, as well as easy manipulation of shapes in the latent space, such as shape interpolation and extrapolation.

On the other hand, deep-learning-based 3D shape generation also poses new challenges. Compared to 2D images, the complexity and irregularity of 3D shapes and the different requirements of practical applications have resulted in the lack of a unified representation for 3D shapes [139]. Instead, various 3D shape representations are used, such as voxels, point clouds, meshes, and implicit functions. Each 3D representation has its own data structure, advantages, and disadvantages. These factors must be taken into account when learning to generate 3D shapes. This survey attempts to comprehensively

[1] BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: Q.-C Xu, quncexu@tsinghua.edu.cn; T.-J. Mu, taijiang@tsinghua.edu.cn(✉)

[2] Department of Computer Science, University of Bath, Bath, UK. Email: Y.-L Yang, y.yang@cs.bath.ac.uk

Table 1 Overview of works on deep 3D shape generation according to shape representations and generator types.

3D Representation	Generator network architecture				
	Encoder-Decoder	Generative Model			
		GAN based	VAE-based	Flow-based	Other
Voxel	[14–20], [21–27]	[28–32], [33–37]	[29, 35, 38–40]	-	[41–44]
Point Cloud	[45–51], [52–56]	[57–64], [56, 65–70]	[45, 71, 72]	[73–77]	[78–82]
Mesh	[83–87], [88–92]	[93–96]	[97, 98]	-	[99–101]
Implicit Representation	[102–106], [107–111]	[112–115]	[116]	[117]	[118, 119]
Structure-based	[120–124]	[125, 126]	[127–132]	-	[133–138]

review existing works according to representation.

For 3D shape generation, the most important component of the deep learning model is usually a generator or decoder to construct 3D shapes from the latent space. It may be coupled with an encoder that maps 3D shapes into the latent space. Unlike discriminative tasks such as shape classification [140, 141] and segmentation [142, 143], 3D shape generation as a generative task is more complicated, as it requires learning a proper distribution in the latent space rather than a feature extractor or discriminator for a specific goal. Also, 3D shapes with good quality and variation are desired when generating and manipulating samples in the latent space. When reviewing existing works according to representation, we also categorize them according to their deep learning models' architecture, particularly the generator.

In the following, in Section 2, we briefly provide a background to 3D shape representations and deep learning network architectures. In Section 3, we analyze prior works according to their 3D representation and network architecture. Then, we provide information on collected datasets used for 3D shape generation in Section 4. We discuss potential future research directions in Section 5, and draw conclusions in Section 6.

2 Geometry and Learning Background

This section introduces 3D geometry learning, the basis of 3D shape generation using deep learning, from both geometric and learning aspects. Interested readers may refer to recent surveys [139, 144] for more detailed reviews. Section 2.1 introduces prevalent shape representations for 3D geometry learning, including voxels, point clouds, meshes, implicit functions, and structure-based representations (see Fig. 1).We present the commonly used neural network architectures for data generation in Section 2.2.

2.1 Shape Representations for Deep Learning

2.1.1 Voxel Representation

The constituent pixels of a 2D image provide a stable and uniform structure for the image. Analogously, voxels with an orderly and regular structure are an intuitive extension of pixels in 3D space: voxel representation is the 3D counterpart of pixel representation. Thus, voxel-based neural network architectures can be straightforwardly constructed by extending image-based structures by expanding the dimensionality of the learning operators such as convolution, pooling, etc. from 2D to 3D [145, 146]. However, the speed of the deep learning model is greatly affected by the dimensionality increase. Therefore, the resolution of the 3D voxel grid is often limited as a trade-off, which can easily introduce step artifacts, affecting the quality of the generated shapes.

2.1.2 Point Cloud Representation

A point cloud is a more accurate representation of 3D shapes than voxel representation. The 3D coordinates of point samples directly represent shape geometry. Their efficient and concise characteristics make point clouds attractive to researchers. Another reason for their popularity is that point clouds are often the raw output format of 3D acquisition devices. Due to the advances in equipment, such as Kinect [1], cost reductions and accuracy improvements of 3D data capture have made point cloud data acquisition no longer an issue. In addition, the massive data and simple data structure of point clouds are relatively favorable for deep learning. However, point clouds lack the characteristics of spatial order and arrangement normally required in deep learning, so instead order-invariant operations like max-pooling as in PointNet [147] or multiple transformation matrices before the convolution operator [148]are needed to compensate.

2.1.3 Mesh Representation

Like point clouds, meshes are another accurate discrete 3D shape representation. Mesh representation further conveys

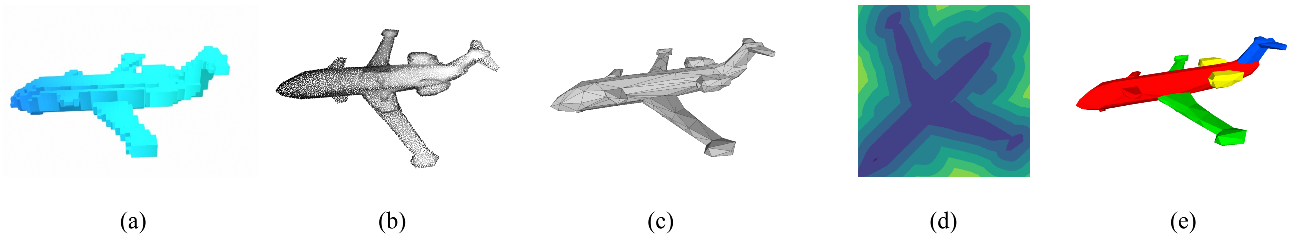


Fig. 1 Different 3D shape representations, using: (a) voxels, (b) a point cloud, (c) a mesh , (d) an implicit representation (signed distance function), (e) a structure-based representation.

information about discrete patches while providing the coordinates of each vertex. The topological connections between mesh vertices recorded by these discrete patches constitute a piece-wise linear approximation of the entire shape surface, serving as a higher-quality 3D shape representation than a point cloud. At the same time, meshes consume less memory than a voxel representation of the same resolution as they only represent the boundary surface rather than the entire volume. However, the irregularity and complexity of mesh topology make it challenging to use for deep learning. Researchers are continuing to explore ways of transforming meshes into a latent vector coding that can be used in deep learning. Recently, various architectures have been developed to learn mesh features by defining basic deep learning operators (e.g. convolution, pooling) based on a specialized mesh structure [149–151].

2.1.4 Implicit Representation

Implicit representations usually rely on implicit functions such as the occupancy function [103] or signed distance function [102] to describe the shape of a 3D model. The neural network learns implicit functions at points and faces that define unique spatial relationships. Implicit representation allow flexible shape topology unlike the fixed topology of a mesh. Moreover, with reasonable memory consumption, implicit representations can increase resolution continuously. However, as the generator output of a deep learning network, an implicit representation cannot reflect the geometric features of the model and usually requires a post-processing stage, like marching cubes [152], to convert it into an explicit shape representation, such as a mesh, before it can be used by downstream tasks.

2.1.5 Structure-based Representation

Structure-based representations decompose a 3D model of a complex shape into a collection of shape primitives. The structure and geometric details of the primitives are usually utilized for training. Structure-based representations pay more attention to high-level structural features between parts of the shape, such as orientation relationships, symmetry, and contact relationships. Compared to the previously mentioned accurate shape representations, using a structure-based representation for 3D shape generation cannot accurately reproduce geometric details but allows better overall shape generation and control.

2.2 Neural Networks for Data Generation

2.2.1 Types

From a 3D generation perspective, we can divide standard neural networks into two different types. The *encoder-decoder* type, mainly like an autoencoder [153], is usually used in a supervised learning task such as generating 3D shapes from inputs in forms such as images or incomplete point clouds. The **generative model** type includes popular models for generation tasks such as GANs [154], and VAEs [155]. Moreover, some models share elements of both of the above types.

2.2.2 Encoder-Decoder

Encoder-decoder networks can be divided into two parts, as implied by the name. First, an encoder typically encodes the input data into a vector in the latent space. As a generator, a decoder ending with deconvolution or fully connected layers is commonly used for shape generation or reconstruction. This type of generator can achieve good performance in a well-trained autoencoder. Most supervised generation tasks, such as 3D reconstruction from a single image, utilize this type of generator.

2.2.3 Generative Models

Among a wide range of generative models, GANs, VAEs, and flow-based models are the most commonly used for 3D shape generation. Here we briefly review the basic ideas of these models, used in the discussion of 3D shape generation in Section 3.

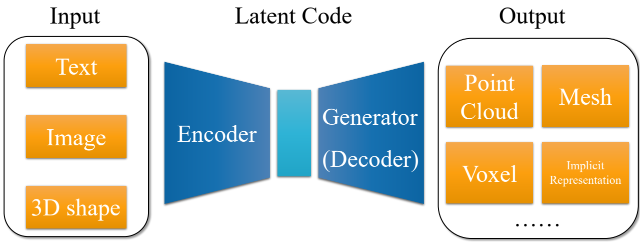


Fig. 2 A deep 3D shape generator is a network that can recover a 3D shape through an encoded vector or under the guidance of a latent vector.

Generative adversarial networks, originally proposed in [154], achieved a significant breakthrough, training a generative model by confronting two networks. A GAN consists of two networks: a discriminator, which estimates the probability that a sample comes from the real data distribution, and a generator, which approximates the real data distribution by trying to fool the discriminator with synthetic samples. The two networks are trained simultaneously via a max-min game. GANs provide a good solution to model data generation posed as an unsupervised problem.

Variational autoencoders [155], another classical generative model, appeared almost simultaneously with GANs. However, while GANs aim to generate new images, VAEs, in addition to this, want to learn an implicit representation and model the distribution of the data. In other words, the dependency from sample variable z to distribution $\mathbf{x}$ in VAE is random, while in GAN, it is deterministic. This makes the inference of VAEs relatively well-defined and the inference of GANs relatively pathological.

Flow-based models differ again. The training process is directly based on maximum likelihood estimation. The classic generative flow model uses a reversible neural network [156–161], and can find a bilateral path not only from the distribution A to distribution B but also from B back to A. Due to the inverse function, there is no need to train an inference model.

As research on generative models for 2D images has progressed, attention has begun to shift from 2D image generation to 3D shape generation. Classical generative network models such as GANs and VAEs are used to generate or restore 3D shapes. However, due to the lack of a unique 3D shape representation, classical generative models encounter different difficulties in generating 3D shapes. This has motivated researchers to explore new methods of 3D model generation. In order to distinguish it from the traditional generative model, from now on in this survey, we say that a network that can recover the 3D shape through an encoded vector or under the guidance of a latent vector is a *deep 3D shape generator*, as shown in Fig. 2.

3 Learning to Generate 3D Shapes

In this section, we comprehensively review use of deep learning to generate 3D shapes. We first categorize existing works based on the 3D shape representation employed, including voxels (Section 3.1), point clouds (Section 3.2), meshes (Section 3.3), implicit representations (Section 3.4), and structure-based representations (Section 3.5). Then we make further classifications depending on the type of generator used for 3D shape generation. A brief summary of generator types and works reviewed in this survey are listed in Table 1. For each representation, we also provide an overview timeline of representative methods.

3.1 Voxel-based Shape Generation

3.1.1 Basics

Voxel representation uses a regular lattice to divide three-dimensional space: the smaller the size of each voxel, the finer the shape details expressed by the voxels. A mesh representation can be extracted from a voxel representation by methods such as marching cubes [152]. The most significant advantage of voxel representation is its regular structure. Since voxels are a straightforward extension of pixels, many deep learning methods used on 2D images can be directly extended to 3D voxel shapes.

3.1.2 Encoder-Decoder

For the supervised 3D shape generation task, many works use customized generators to generate or reconstruct 3D shapes in voxel representation from the latent embedding of 2D images. Using the regular structure of voxel representation, the generators in [13, 15, 16] are based on 3D convolutional neural networks for predicting geometric shapes.

In early work, Yan et al. [14] introduced a perspective transformer net that can generate volumetric 3D shapes from images without 3D supervision. First, the input image is coded into the latent unit as a $1 \times 1 \times 512$ vector by a convolutional encoder. Then, in the decoder, a volume generator recovers the shape at $32 \times 32 \times 32$ size. The perspective transformer module then projects a 3D sample to a 2D silhouette for reconstruction supervision. Meanwhile, Girdhar et al. [15] proposed a baseline work for 3D shape generation from image priors, called a TL-embedding network (see Fig. 4). The T-net, a 3D convolution-based autoencoder, learns both the 3D shape embedding feature and the 2D embedding feature of their rendered images. The L-net uses the 2D encoder and 3D decoder to infer the output for input images. Euclidean distance loss is applied to align the embedding vectors from

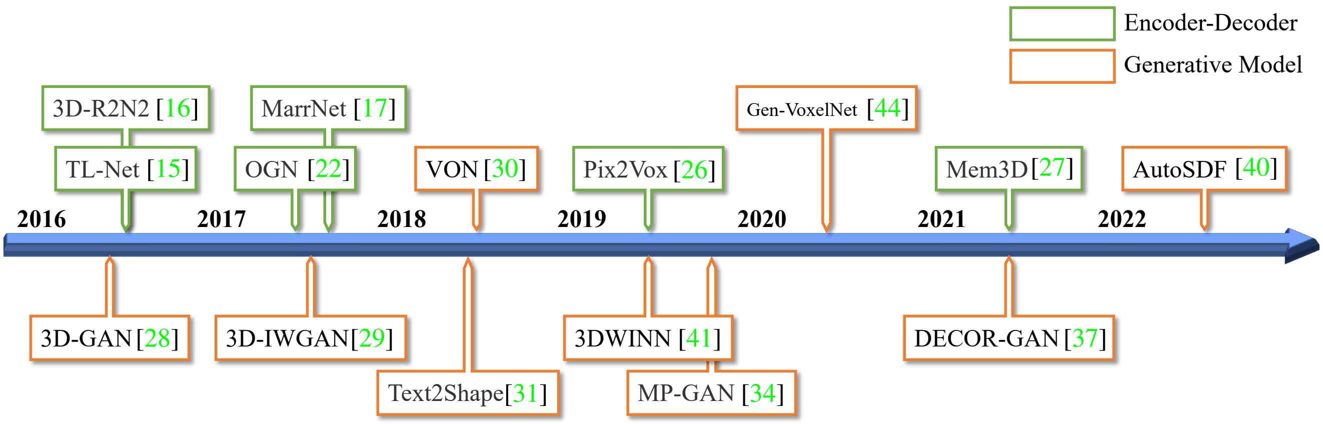


Fig. 3 Timeline of shape generation methods based on voxel representation.

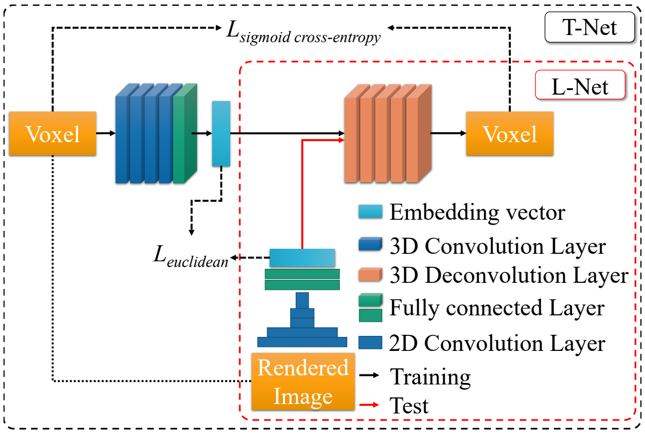


Fig. 4 Architecture of TL-NET for 3D voxel generation (from [15]). Input and output are 20 × 20 × 20 voxel grids. In T-net training, images are rendered based on the input 3D models. In L-net testing, images come from real data.

the corresponding 3D shape and 2D image. Furthermore, sigmoid cross-entropy loss is applied for reconstruction supervision. The 3D Recurrent Reconstruction Neural Network (3D-R2N2) [16], another prevalent baseline method for 3D voxel generation, applies the standard long short term memory (LSTM) mechanism as a 3D convolutional LSTM between the 2D-CNN encoder and 3D deconvolution based decoder (see Fig. 5).

Generic shape generation methods like MarrNet [17] go through a 2.5D sketch to decouple input into normal, depth, and silhouette. Then an autoencoder is used to estimate shapes from the above information. Based on MarrNet, ShapeHD [18] introduces a naturalness loss with an adversarially pre-trained convolutional net as a discriminator to improve generation quality. Another derivative of MarrNet is GenRe [19], which considers generalizing the model to unseen categories by decoupling geometric projections from shapes. Using multi-view images (i.e. images from multiple viewpoints) can bring more priors to help shape generation; a learnt stereo machine architecture was proposed by Kar et al. [20]. Liu et al. [21] proposed the Variational Shape Learner (VSL) as an unsupervised 3D voxel generative model built on Neural Statistician [162]. This model can extract expressive features and generate 3D shapes from sampled latent vectors. The shape encoder with skip-connections learns global latent features while the shape decoder inverts the encoder and translates latent features back to voxel-based 3D shapes.

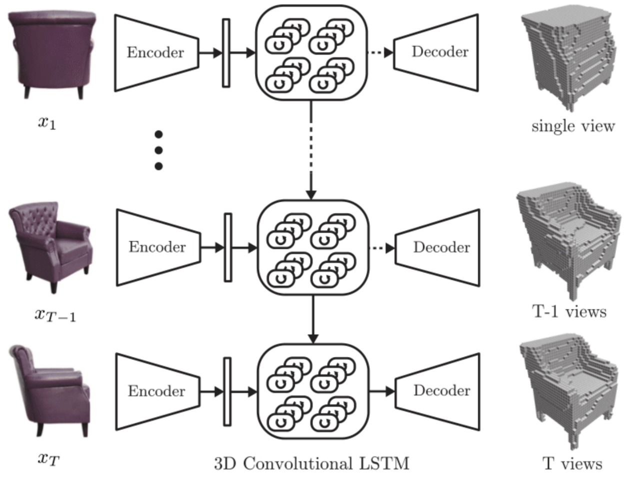
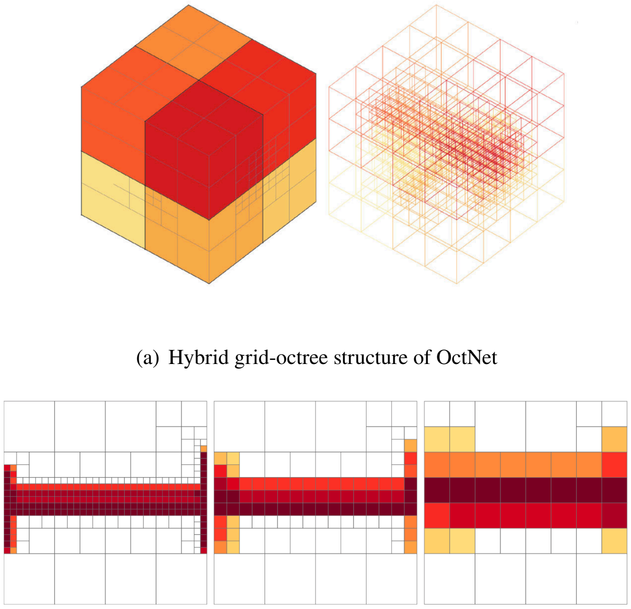


Fig. 5 Overview of 3D-R2N2 [16]. The network takes as input a sequence of images from arbitrary viewpoints and produces a voxelized 3D reconstruction. Image courtesy of [16].

To reduce memory consumption, octree generative networks (OGN) [22] adopt a hierarchical, memory-efficient octree data structure to generate high-resolution voxels with an up-convolution decoder built from three fully connected layers. Häne et al. [23] reconstruct 3D shapes in steps with a coarse-to-fine approach. An octree structure is adopted here for high-resolution voxel generation.



(a) Hybrid grid-octree structure of OctNet

(b) Spatial structures in different layers of OctNet

Fig. 6 OctNet [163] learns a voxel representation at high resolution with low memory consumption. Images courtesy of [163].

Following efficient octree-based learning methods [163, 164] (see Fig. 6), a two-stage 3D-CFCN (cascaded fully convolutional network) architecture was proposed by Cao et al. [24]. Later, Liu et al. [25] presented a coarse-to-fine approach for generating high-fidelity volumetric 3D shapes from noisy and incomplete input. Following Cao et al., octree-based learning [22, 163, 164] was adopted as the backbone in [25] to reconstruct high-quality shapes. A raw RGB-D scan input is used to fuse a low-resolution truncated signed distance function (TSDF) volume with color. In the first 3D-FCN, a structure-related feature is learned to guide the second 3D-FCN, which accepts a high-resolution TSDF and color as input and generates high-resolution patches. Generation from two different resolutions of 3D-FCN can better handle the global and local features of the 3D shape.

Unlike the above octree-based frameworks, Pix2vox [26] adopts a coarse-to-fine approach to generate a voxel-based shape from single or multi-view images. Decoders with five 3D transposed convolutional layers generate shapes for different input latent codes and fuse them to produce a coarse shape, which is refined by a U-Net like 3D convolutional encoder-decoder module.

Recently, Yang et al. [27] investigated how to use shape priors and hidden information from images to help generate better 3D shapes. A robust method called Mem3D that generates a 3D shape from an occluded or noisy background image was proposed. With learnt shape priors structured in 'image-voxel' pairs, Mem3D comprises not only a traditional image encoder and a 3D volumetric decoder but also a memory network and an LSTM shape encoder. The memory network helps retrieve the closest 3D volumes to the input image and send them to the LSTM shape encoder. The shape encoder converts shapes into a vector with a shape prior using the LSTM mechanism. Such a prior vector concatenated with the image feature can offer hidden information about the unseen part in the image to improve the generated results.

3.1.3 Generative Models

3D-GAN [28] and 3D-IWGAN [29] are pioneers for unsupervised 3D voxel shape generation, using generative adversarial networks (GANs) which extend 2D image generation approaches (see Fig. 7).

3D-GAN [28] generates voxel grids of size 64 × 64 × 64 from sampled Gaussian noise while the generator is designed by referring to [165]. The cross-entropy loss used in the classic GAN model is the main objective for the generator to synthesize the shape while the discriminator is adapted from 2D. Smith et al. [29] proposed 3D-VAE-IWGAN to improve generated results. Moreover, they used fully connected layers with 2048 nodes instead of the sigmoid layer in the architecture of 3D-GAN [28].

Meanwhile, Brock et al. [38] proposed a voxel-based VAE architecture to learn 3D shapes. The encoder network consists of 4 convolution layers with a fully-connected layer while the decoder is duplicated but inverted. Inspired by ResNet [166], Brock et al. introduced Voxception-ResNet to improve performance. Also, in the VAE-based framework, Balashova et al. [39] brought a structure-aware loss with a pre-trained structure decoder to enhance generation quality.

Following 3D-GAN [28], the Visual Object Network (VON) [30] improves the 3D generation results by use of Wasserstein distance from WGAN-GP [167, 168], together with a 2D texture network with 2.5D sketches through differentiable projection. Chen et al. [31] proposed Text2Shape, an architecture generating 3D voxel shapes from text input. A joint representation learning approach is employed for the cross-modality learning task. In addition, the CWGAN framework is used for conditionally generating colored shapes following input text. Knyaz et al. [32] proposed a pipeline to recover 3D shape from a single 2D image through an adopted z-GAN architecture [169] and frustum voxel model. This provides precise alignment between voxel slices and image contours. Huang et al. [41] presented 3DWINN based on introspective neural networks [170] and involved Frechet inception distance in module evaluation.

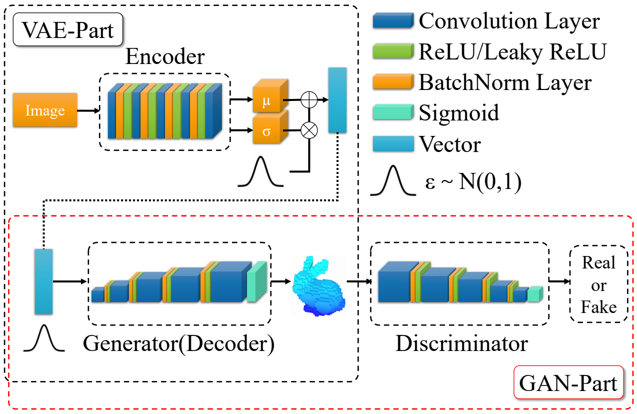


Fig. 7 Typical 3D VAE-GAN architecture proposed in [28] and improved by [29]. In the VAE, the encoder consists of 2D spatial convolution layers with ReLU and batch normalization layers. 3D volumetric convolution layers are used in the generator and discriminator instead of 2D convolution. Unlike simple GANs, VAE-GANs use the sampled latent representation as input to the GAN.

Following Projective GANs [33], which involve 2D projection into 3D GAN training, Li et al. [34] proposed the Multi-Projection GAN (MP-GAN) for voxel-based shape generation. Differential projection is defined as a non-linear operator from a high to low dimensional distribution. The generator is trained to compete with multiple discriminators, assessing where the projection comes from. The view prediction network with cluster module is employed to iteratively alternate training with MP-GAN. Note that silhouettes as projections from multiple views are not required to have correspondences with the known viewpoints.

Unlike the traditional 3D GAN framework, Khan et al. [35] proposed a generative model involving primitive compositions which can be controlled with suitable interpretability. Before generating complete shapes, it uses a primitive GAN to generate a parsimonious shape configured by primitives. A VAE is used to connect the primitive GAN and 3D GAN by encoding the parametric primitives into latent space used for sampling. This model can be jointly trained on all the categories in the dataset.

Henzler et al. [36] introduced PLATONICGAN, which can generate 3D volumetric shapes from an unstructured collection of images while involving a rendering layer in their network architecture. Like other works, the image input is encoded into a latent vector by a generalized 2D feature extractor. Then, the generator accepts a latent vector and generates a 3D voxel shape. The rendering layer accepts the generated 3D voxel with view sampling and additional information like visual hull, absorption-only, and emission-absorption from the image formation model. Finally, a rendered 2D image is fed into the discriminator to complete the full generative model training.

Other works are based on an autoregressive model for voxel-based shape generation. The Octree Transformer [42] uses an autoregressive generation method with an octree-based network. With the transformer involved, the input is encoded into a sequenced octree. Then, compressed shorter sequence latent vectors can be trained with a classic transformer decoder. This work shows that the proposed compression scheme can help reduce the sequence length using the octree data structure and still be compatible with autoregressive generation. Also employing the autoregressive model is AutoSDF [40]. While VQ-VAE [171] learns quantized and compact latent representations for images and Esser et al.[172] learned autoregressive generation from discrete VQ-VAE, AutoSDF extends the method of Esser et al. to the domain of 3D shapes but with a generic non-sequential autoregressive prior. This method can generate shapes from image and text as conditions through ResNet or BERT [173] as the encoder, respectively.

DECOR-GAN[37] is a state-of-the-art method for detailed voxel generation. For the generator, it utilizes a 3D CNN, while for the discriminator, it uses 3D Patch-GANs [169] with a receptive field of $18 \times 18 \times 18$. While it is possible to reduce memory consumption based on various space partitioning techniques, these approaches have complex implementations, and existing data-adaptive algorithms are still limited to relatively small voxel grids.

In addition, some works use energy-based models (EBMs) [174] for voxel-based shape generation. with 3D DescriptorNet [43] and Generative VoxelNet [44] achieving state-of-the-art results. By associating VoxNet [146] and [175], 3D DescriptorNet describes 3D shapes via a probability density function. It synthesizes and generates new shapes through a Langevin dynamics approach to sample the distribution. As an extension of 3D DescriptorNet [43], Generative VoxelNet [44] adopts a multi-scale energy-based generative model for high resolution 3D shape synthesis and generation.

3.1.4 Summary

Based on direct extension of image generation, using voxels for shape generation gives researchers the benefit of the regular structure of voxels. Voxel-based shape generators are analogous to pixel-based image generators in terms of overall structure and loss functions. The regular representation makes it easy for the generator to output shapes as a 3D volumetric grid data. To generate the final results, operations like up-sampling and up-convolution similar to those in a 3D CNN network are used in most works. However, such shape generators require a large amount of memory due to

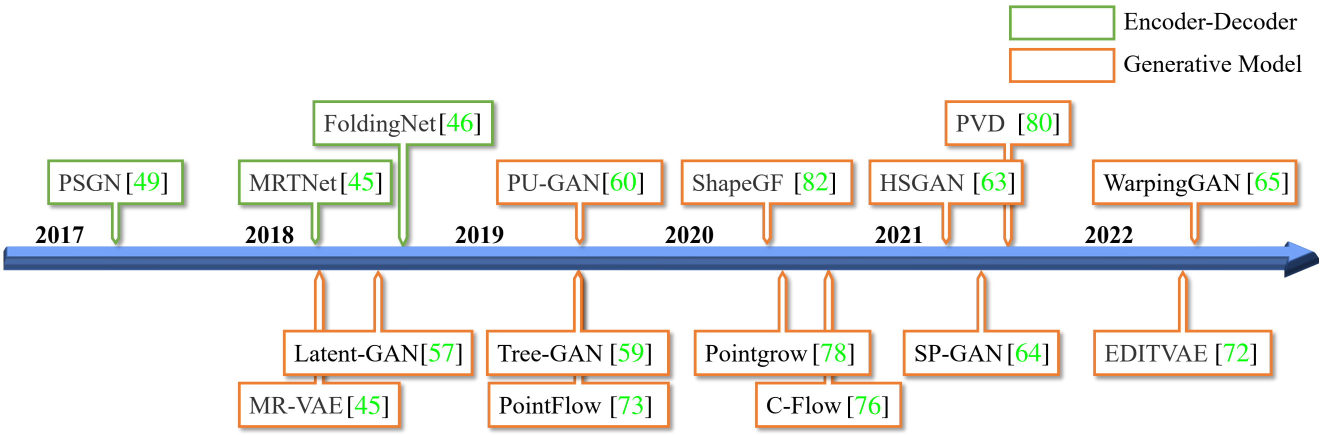


Fig. 8 Timeline of shape generation methods based on point clouds.

the increased dimensionality, so generating high-quality 3D shapes is challenging. How to reduce the memory cost is still an important research issue.

3.2 Point-Cloud-based Shape Generation

3.2.1 Basics

In favor of using point clouds for 3D shape generation are the conciseness of their geometry primitives, the simplicity of the overall structure, and the convenience of 3D acquisition. The main challenge is that point clouds lack a regular structure so cannot easily fit into convolutional network architectures that exploit spatial regularity.

3.2.2 Encoder-Decoder

While irregularity and disorder characterise point clouds, how to learn from such unstructured data has rapidly become a topic of interest in geometric learning. As pioneering work, PointNet [147] proposed the use of multi-layer perceptrons (MLPs) and symmetric functions like max-pooling to extract features directly from the point set. Pointnet++ [176] adopted U-Net and a hierarchical network structure as better adapted to segmentation and classification tasks. Later, PointCNN [148] was introduced with a convolutional operator, and Wang et al. [177] involved a dynamic graph CNN by employing the EdgeConv operator. Recently, the Transformer [178] was first introduced to point cloud processing by PCT [141], starting the fashion for using an attention model for point cloud representation learning [179, 180].

After these point cloud learning methods were proposed, many works have considered how to efficiently encode and decode point clouds, especially how to generate, reconstruct or recover point cloud shapes from partial inputs (e.g. images and incomplete point clouds), as a supervised learning task.

The early MRTNet [45] introduced multi-resolution convolution into the encoder and decoder architecture like PointNet++. For shape generation, simple modification with additional VAE loss allows the generation of point clouds from sampled latent codes. FoldingNet [46] proposed a folding-style operation to generate point clouds by learning mapping functions. Recently, an adversarial autoencoder was employed in 3DAAE [47] for generating point clouds. A discriminator is used to distinguish the sample generated. Furthermore, shape interpolation and geometric arithmetic are achievable using this framework.

A deformation-based method can also be used for point cloud generation. DeformNet [48] proposed to generate topology-preserving 3D shapes from single-view images through free-form deformation (FFD) [181]. The initial template shape is retrieved from a CAD model dataset. The 2D CNN encoder and the 3D CNN encoder encode the image and the initial template into a joint latent code. Then, the decoder produces a deformation field for FFD layers to move the point cloud following the output prediction to generate the final point cloud.

To generate point clouds from images, customized generators based on PointNet and its variants [176] were rapidly produced. The Point Set Generation Network (PSGN) [49], as one of the pioneers utilizing a conditional shape sampler, can predict multiple reasonable 3D point clouds from an input image (see Fig. 9). Dual predictors are applied in their generator to adapt to the large and smooth surface.

Recent works like Wei et al. [50] generate a point cloud shape with a partially supervised generative network limited using random input. A frontal constraint is adopted in single-view training to force the model to pay attention to the front part, while a diversity constraint simultaneously controls the

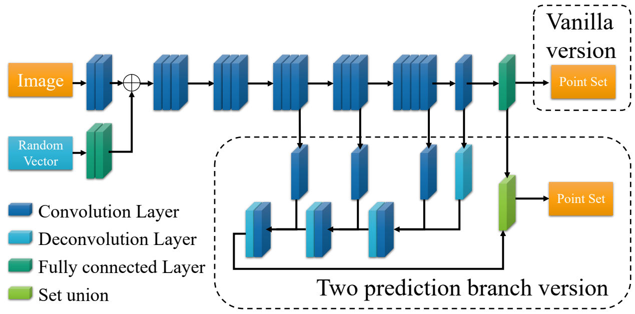


Fig. 9 Point Set Generation Network (PSGN) [49], a pioneer of point cloud generation. Improving the basic version involves branches with deconvolution and fully-connected layers. Skip connections are used to boost performance. Set union is applied for merging predictions.

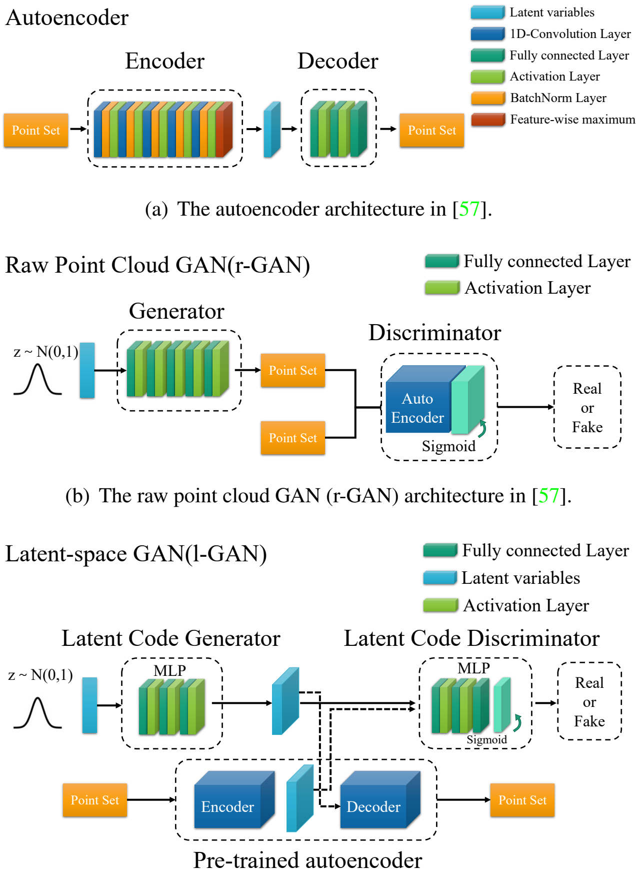
diversity of generation. This model can generate multiple plausible shapes from a different view during inference. Going further, Hu et al. [51] proposed a complete pipeline to generate a densely textured 3D point cloud from a single image.

Other works [182, 183] based on point cloud rendering have appeared in recent years. Lin et al. [52] use an image encoder to map the input to the latent space and employ a pseudo-rendering method for joint 2D projection optimization. The generated 3D structures are fused with the final point clouds from various viewpoints using a 2D convolution-based structure generator. Insafutdinov et al. [53] learn both shape and camera pose from two different input views with a differentiable point cloud represented by density functions. Later, Chen et al. [54] proposed a method generating a 3D point cloud by 2D projection matching without 3D supervision. Unlike other differentiable rendering-based works like [53], Chen et al. [54] abandon per-pixel difference but chooses sample point supervision from ground-truth silhouettes.

Komarichev et al. [55] proposed a method that generates point clouds via joint learning. First, it learns a joint latent embedding from input point clouds and image sets. Then, a geometry-aware autoencoder encodes the point clouds. The latent variable comes from the autoencoder, so they propose a mixer network to map them into a joint latent space. Then, a joint generative model is applied to generate a joint latent vector which can be employed for several multi-modal shape generation tasks to recover the latent vectors of the image and point cloud separately. Finally, they obtain the point cloud and the image from decoders corresponding to each embedding.

3.2.3 Generative Models

Point cloud generation based on GANs has become an active topic in recent years. Achlioptas et al. [57] proposed the first work to use a deep generative model to generate point cloud shapes from sampled Gaussian noise vectors with r-GAN or latent-GAN (see Fig. 10). Valsesia et al. [58] improved r-GAN [57] by using graph convolution to achieve better generation results. Shu et al. [59] introduced another convolution architecture called Tree-GAN using a tree-structure for point cloud learning without prior-like connectivity in a graph [58].



(a) The autoencoder architecture in [57].

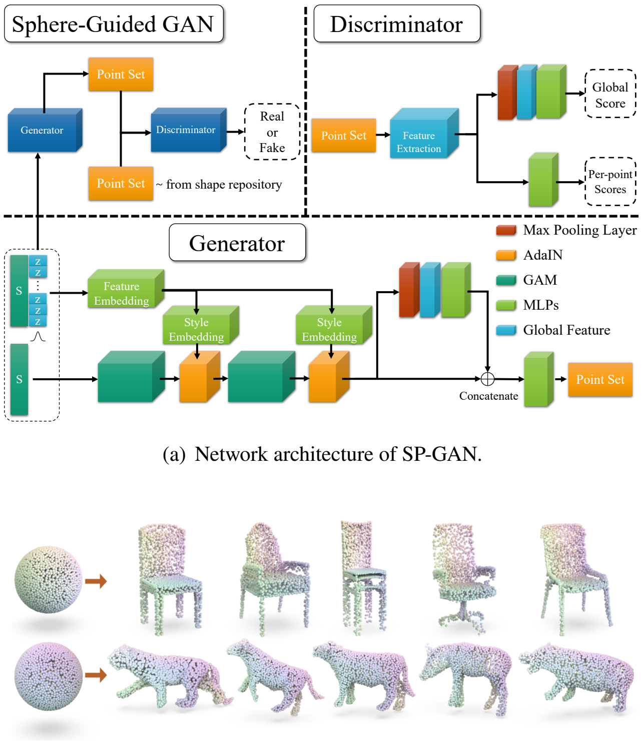
(b) The raw point cloud GAN (r-GAN) architecture in [57].

(c) The latent space GAN (l-GAN) architecture in [57].

Fig. 10 Three different architectures proposed in [57]. (a) The autoencoder adopts PointNet with 1-D convolution as the encoder. Feature-wise maximum produces the k-dimensional vector for latent space. (b) r-GAN and (c) l-GAN are used as baselines in much following 3D shape generation work, not limited to point clouds.

PU-GAN [60] uses an upsampling technique to generate a dense point cloud from sparse input. In the generator, in addition to feature extractors, it also expands features, reconstructs coordinates, and applies farthest sampling to ensure a dense result.

Later methodologies based on GANs proposed different technical solutions. To generate high-resolution point clouds, a 3D generative model applied in the spectral domain was proposed by Spectral-GAN [61]. Unlike a spatial GAN, it treats point cloud data as spherical harmonic moment vectors (SMVs) that encode points into the structure and fixed dimension vectors with highly correlated relationships between elements. This feature makes learning spectral.



(a) Network architecture of SP-GAN.

(b) Point cloud generation results from SP-GAN.

Fig. 11 SP-GAN [64] leverages a guiding sphere to generate point cloud shapes with different geometry and topology. (a) Network architecture; S is the guiding sphere vector and z is sampled noise from a Gaussian distribution. Feature embedding and style embedding are implemented using MLPs. GAM stands for Graph Attention Module and AdaIN denotes Adaptive Instance Normalization. (b) Outputs, courtesy of [64].

SAPCGAN [62] involves a self-attention mechanism in the point GAN. A graph aggregation layer fuses a binary tree framework and self-attention for precise point cloud data generation. Li et al. [63] proposed HSGAN that incorporates a graph convolution network with a self-attention mechanism that can simultaneously learn local geometry and global topology. SP-GAN [64] has a similar generator structure to Style-GAN [184] and achieves plausible results (see Fig. 11). This work binds sampled noise to a template sphere, thus solving the point cloud discontinuity problem. With the graph attention module based on DGCNN [177] and the adaptive instance normalization module as the feature encoding extraction network, feature embedding is combined with hierarchical features through style embedding. The coordinate information of the 3D point cloud is recovered from the feature map through a max-pooling layer and MLPs in the last phase with exact iterations. Compared to previous work, SP-GAN can generate point clouds with less noise and more detail. The model implicitly embeds dense and consistent correspondences between generated shapes.

After pointing out challenges faced by FoldingNet [46] and AtlasNet [83], Tang et al. [65] introduced WarpingGAN which includes code enhancement and unified local-warping modules. In the generator of WarpingGAN, code enhancement uses MLPs to change the input latent code into a global shape code. The local code (split from the global code) concatenated with the shape prior and the global feature is sent into the unified local-warping module to predict points by MLPs. This unified local-warping mechanism can conditionally warp the uniform distribution of 3D priors into various local shape regions, making the generation process more effective and efficient.

In recent years, progressive methods have also been widely employed in 3D point generation. Hui et al. [66] developed a progressive deconvolution network for point cloud generation. It maps the latent vector to a high-dimensional feature space. A constructed deconvolution operation is proposed to use the similarity between points and interpolation to enlarge the feature maps. Classic MLPs are applied to generate locations in point clouds after the deconvolution network. While the conditional GAN idea was applied in the progressive model PCGAN [67], another progressive point cloud generation method is based on the dual-generator framework introduced by Wen et al. [68]. While both generators share the same discriminator, the first generator sends upsampled dense results into the second generator to refine the rough shape, like a noise filter.

Similarly, an autoregressive method called Pointgrow [78] was proposed to predict the coordinate distribution of a 3D point cloud from the training set. A deep neural network takes previously generated values as input and outputs a distribution of the values currently under consideration. When generating points, the points are sampled one by one according to the estimated probability distribution. Due to the inherently iterative approach of autoregressive models, size of the point cloud's cannot be readily changed.

Based on the classic generative flow-based model, PointFlow [73] proposed a VAE-based point cloud generation method with continuously normalizing flows (see Fig. 12). Usually, flow-based generative methods learn to model the distribution of points in a shape through reversible parametric transformations of points. PointFlow generates point clouds from a standard 3D Gaussian prior based on continuously normalizing flows. Discrete normalizing flows with affine coupling layers generate point clouds in DPF-Net [74]. Following the same training framework as PointFlow, SoftPointFlow [75] generates point clouds with high-quality details due to the ability to capture innate manifold structure. Pumarola

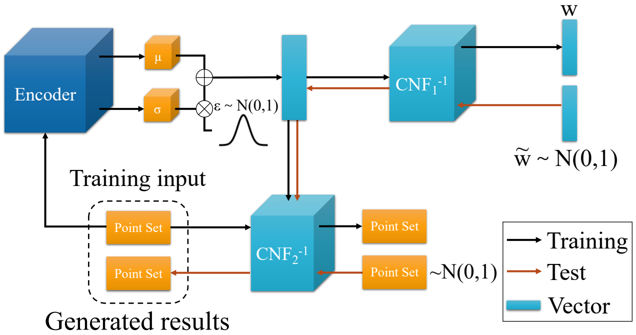


Fig. 12 Architecture of PointFlow [73]. The point encoder is adopted from [57]. Two continuous normalizing flows (CNF) are trained with variational inference. CNF$_1$ is used for modeling distribution priors, while CNF$_2$ focuses on modeling the reconstruction likelihood as the decoder.

et al. [76] proposed a parallel conditional flow scheme called C-Flow in a DNF-based architecture like DPF-Net. Mixed normalized flow was introduced by Postels et al. [77] which shows better shape generation results than single flow models.

Like flow-based models, energy-based methods have been favored by researchers recently. Luo et al. [79] used an energy-based model, a diffusion probabilistic model as in Sohl-Dickstein et al. [185], to achieve conditional 3D shape generation with an encoder for latent shape. Compared to other flow-based models [73, 74], Luo et al. [79] adopted a reverse diffusion Markov chain to model the distribution of point clouds; it does not require invertibility or assume order. On the other hand, flow-based models often treat 3D shape generation as a probabilistic problem and solve it by sampling and moving these points based on a learned distribution transformation. Differing from Luo et al. [79], Zhou et al. [80] proposed another denoising diffusion probabilistic model with hybrid point-voxel representation to generate high-quality 3D shapes by filtering Gaussian noise. Meanwhile, Xie et al. [81] proposed an energy-based model for point clouds which learns each point's coordinate coding and fuses them to give a global energy for the whole point cloud. Furthermore, this is also the first generative model that provides an explicit density function for unstructured point cloud data. The learned short-run Markov chain Monte Carlo (MCMC) method can generate a diverse point cloud, reconstruct incomplete shapes, and interpolate between point clouds.

Besides GAN-based architectures, the VAE is also a prevalent generative model in point cloud processing [45, 71, 72]. Gadelha et al. [45] proposed a multi-resolution tree structure for point cloud learning called MRTNet. In detail, the encoder and decoder consist of a sequence of multiple resolution convolution blocks that receive input from each resolution that can be upsampled, pooled, or concatenated. As a result, MR-VAE is easily implemented and generates diverse point clouds by sampling latent space. Kim et al. [71] introduced SetVAE, which applies set transforms to the classic VAE model. Since SetVAE learns the latent variable at various scales by introducing top-down latent dependency and novel bottleneck equivariant layers, multi-scale structure features can be easily captured to build hierarchical data. Overall, SetVAE generates diverse, high-quality point clouds with fewer parameters. Recently, Li et al. [72] proposed EDITVAE, which learns decoupled latent information that is linearly mapped from the global latent information and generates diverse part-aware point sets by simply sampling a Gaussian distribution.

In addition, there is some part-aware research into point cloud generation like [56, 69, 70, 72, 121]. Mo et al. [69] included the part-tree structure in the point cloud learning PT2PC (part tree to point cloud) framework. Their conditional GAN model is composed of a part-tree conditional generator and discriminator. Besides the part-tree encoder and decoder, point cloud decoding is also processed in the generator. The generated points are encoded to form part-tree structures for discriminator judgement. Gal et al. [56] proposed MRGAN for part decoupling and adopt an AdaIN layer, similar to StyleGAN [7], for each root node when shape generating, to achieve part-control. Yang et al. [70] proposed CPCGAN for controllable point cloud generation; it consists of a structure GAN for middle-level point clouds with semantic labeling and a final GAN for the complete point cloud.

Slightly differing from the above work, Cai et al. [82] learned to use a gradient field to generate a point cloud from an arbitrary prior point cloud. In addition, an extended score-based method is used to learn the conditional distribution. They treat 3D points as a distribution and use a neural network to model the gradient of the log-density. The latent-GAN proposed by Achlioptas et al. [57] is employed to learn the distribution of the input latent code. Like other probabilistic methods, additional training is required for the autoencoder in a two-stage training process.

3.2.4 *Summary*

It is not difficult to see that in the classic model of point cloud generation, the generator is likely to be the PointNet-based decoder architecture, which often generates a fixed sequence of 3D point coordinates. Chamfer distance and Earth mover distance are the most commonly used metrics to evaluate point cloud differences. Thanks to PointNet [147] and PointNet++ [176], fully connected and deconvolution layers are the most common methods for generators to produce an unorganized point sequence.

3.3 Mesh-based Shape Generation

3.3.1 Basics

As the most popular 3D shape representation in computer graphics, meshes contain not only 3D surface geometry but also topological information. However, the non-canonical mesh structure with irregular topological connections and no regular parametric domain causes difficulties for mesh-based deep learning and generation tasks.

3.3.2 Encoder-Decoder

To simplify the problem, the mesh surface can be parameterized onto a 2D canonical domain so that classic CNN architectures can be adopted for the parameterized shape which has a regular structure. Depending on the topology of the mesh, parameterization is usually performed in a 2D plane (for disk-like shapes) or onto a sphere (for sphere-like shapes).

Using parameterization in the 2D plane, one can implement encoder-decoder architectures using standard 2D convolution operations. For a spherical parameter domain, one has to use spherical convolutions. Geometry images and spherical parameterizations are the most commonly used mesh parameterization techniques [186, 187]. They are, however, suitable only for genus-0 meshes with disk- or sphere-like topology. Meshes of arbitrary topology need to be cut into disk-like patches and then unfolded onto the 2D plane [188]. Finding the optimal cut for a given surface mesh, and more importantly, finding cuts that are consistent across shapes within the same category is particularly challenging. Naively creating independent patches using geometry images for a shape category and feeding them into deep neural networks can easily fail to generate coherent 3D shapes.

Since a mesh is composed of vertices, edges, and faces, some works treat the mesh as a graph, inspired by spectral learning [189–192]. In Mesh-CNN [149], to resolve the ambiguity that the one ring neighbors of an edge can present the triangles in different vertex orders, the authors design edge features as descriptors that are invariant to similarity transformations. The pooling operator collapses edges for feature aggregation to achieve a similar effect to pooling in 2D CNNs. Feng et al. [193] proposed MeshNet for deep learning using mesh representation. While initial values of faces are used to learn spatial and structural features, neighboring information is aggregated and fed into the Mesh Conv block. Then, a pooling function is applied to these features to generate global features for downstream tasks. Recently, a mesh representation learning method was introduced by Liu et al. [194] which is based on a subdivision process to change mesh geometric features. Later, based on the mesh subdivision operation, Hu et al. [151] proposed SubdivNet, with a novel convolution style operating on mesh faces directly; convolution and pooling operators are defined analogously to those for images. The convolution kernel operates on the mesh surface and pools the faces from 4 to 1; it assumes the surface has been remeshed into a Loop subdivision structure. To achieve efficient neighbor indexing for convolutional operations on a graph-like mesh data structure, the re-index operator from Jittor [195], a high-performance deep learning framework for geometry, is adopted as a flexible solution. Such an analogy enables a pyramid structure similar to an image convolution network, resulting in very good results in 3D geometry learning tasks.

In terms of encoder-decoder-based mesh generation, the early AtlasNet [83] generates surface patches to cover the entire 3D shape. Image2Mesh [84] attempts to infer 3D shape through 2D images by learned priors with free-form deformation (FFD). Other works like Pixel2mesh [85, 86] investigate 3D shape generation from 2D images using mesh deformation based on a graph-based convolution network. The key part is a deformation network to deform a template mesh (usually an ellipsoid) for target shape generation (see Fig. 14). The multi-layer structures of the deformation network cause the complete deformation phase to operate from coarse to fine. Instead of producing vertex positions directly, deformation-based methods like [85, 86] generate vertex position offsets. Together with geometric regularization based on Laplacian loss, this strategy makes vertex positions change smoothly so as to avoid self-intersections.

To overcome topological restrictions, Pan et al. [87] proposed a topology-adaptive framework to help reduce artifacts in mesh generation arising from topology. Their method mainly relies on mesh deformation but has an additional topology modification module that adapts intermediate topology according to the input image. Shi et al. [88] focus on mesh structure and parts, proposing a complete pipeline consisting of a geometry structure extractor, a geometry-aware mesh deformation module, and a fine-grained mesh editing module. In another approach, Tang et al. [89] proposed a skeleton-bridged method to learn 3D shapes with complex topology in mesh representation. First, parallel MLPs are used to infer key skeleton points. Then a base mesh is generated from a coarse volume built on these inferred key points from 3D CNN. Finally, a graph convolution network optimizes the vertices to produce the final mesh based on the encoded input image. Meanwhile, Gkioxari et al. [90] proposed Mesh R-CNN, which adopted Mask R-CNN [196] with 3D shape

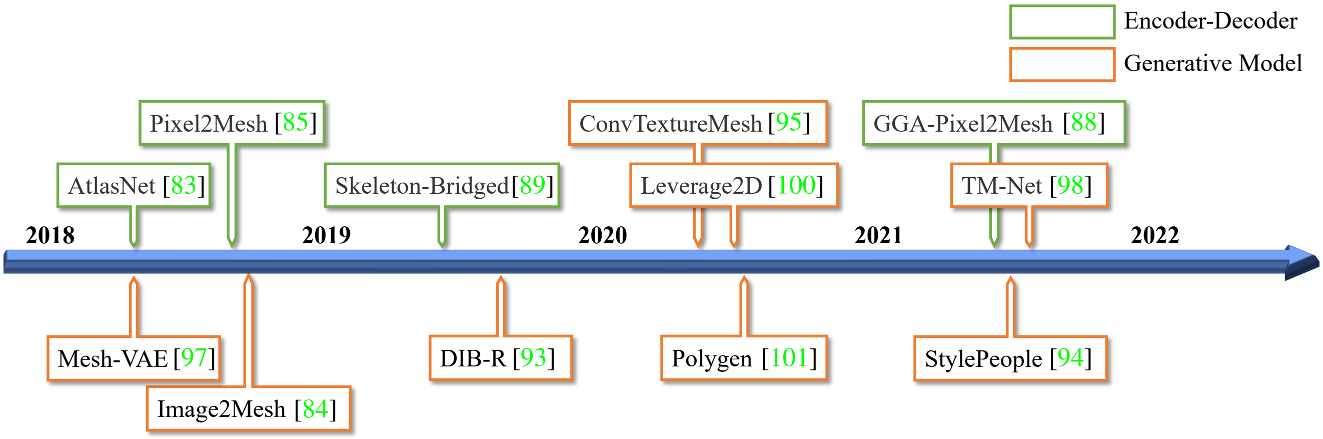


Fig. 13 Timeline of shape generation methods based on mesh representation.

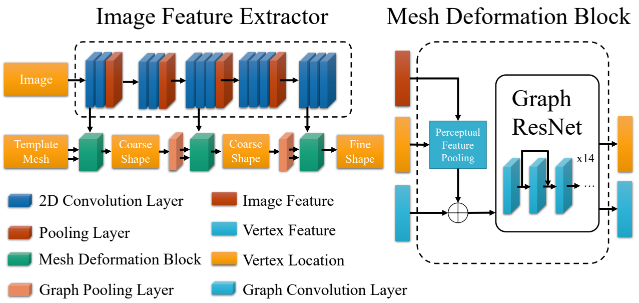


Fig. 14 Network architecture of [85]. A graph convolution network module is used in the deformation block.

inference. A cube-based mesh is produced from predicted coarse voxels, and vertex alignment with graph convolution is applied later to refine the initial mesh. Hui et al. [91] presented the DT-Net framework, which can generate meshes with flexible topology. First, taking an image or voxel as input, the topology formation module learns a function that maps the latent input code to the template model with the correct topology. Then, the technique in BSP-Net [137] is adopted to assemble the implicit field of the template. After both implicit and explicit templates have been produced, training is performed with occupancy from the sampled ground truth shape, while inference progressively applies deformation with condition codes on the explicit template.

Unlike other 2D to 3D works, Zhang et al. [92] focused on generating shapes from sketches. A viewpoint is also provided as additional input to help generation. The convolution encoder decomposes image features into view space and shape space. The decoder produces vertex offsets used for template deformation. Differentiable rendering and discriminator losses are used to increase the quality of generated shapes.

3.3.3 *Generative Models*

Various outstanding works have appeared in recent years based on use of generative models for the generation of shapes based on mesh representation.

Early on, Jimenez et al. [99] adopted a sequential generative architecture extended from [197] to generate 3D volumes or meshes. Variational autoencoders (VAEs) were later applied for mesh model deformation to synthesize new shapes in [97]. After differentiable rendering received great interest for 3D shape reconstruction and generation, research started to replace 3D supervision with 2D supervision [198, 199]. Chen et al. [93] introduced the Differentiable Interpolation-based Renderer (DIB-R) and proposes a framework for adversarial 3D object generation. This was the first generator to learn shape and texture simultaneously.

A deformation-based method for human avatar generation called StylePeople was proposed by Grigorev et al. [94], which involves adversarial learning and neural rendering. StyleGANv2 [184] is used to produce the neural textures that implicitly encode non-modelled geometry like hair and clothing. The deformable body meshes based on SMPL-X [200] are provided with the generated textures and rendered by the neural renderer. Finally, the adversarial module composed of three different discriminators is used to judge the output identity.

Henderson et al. [100] proposed a generative model that can generate meshes with texture. First, an encoder predicts the posterior distribution of the latent variable from the input image. Then, a shape decoder with a color decoder generates vertex positions and face colors. A differentiable renderer renders the final image, which is supervised by the image reconstruction loss. In addition, mesh parameterization is also utilized to ensure that the generated mesh does not contain self-intersection. Pavllo et al. [95] proposed a convolution mesh representation, a displacement map used for template mesh deformation. This is compatible with 2D convolution architectures for both the mesh and its textures. Together with a GAN framework, it can produce a textured 3D mesh from a 2D image. While the convolution-based generator always produces full textures, the generated displacement map and texture are concatenated and sent to discriminators. Differing from DIB-R [93], Pavllo et al. [95] used a pose-independent representation converted from the convolution mesh in a 2D GAN model to reduce issues caused by the pose. This work also pioneered generation of 3D meshes from text conditions. Later, Pavllo et al. [96] proposed a new pipeline to generate meshes without keypoint annotations with the same generative architectures as in [95].

Polygen [101] treats mesh generation as an autoregressive sequence modeling process. They divide mesh vertices and faces into different parts and apply the transformer mechanism to model mesh vertex and face sequences with flexible and variable lengths. They point out that the transformer's ability to aggregate information from parts can handle object symmetries and non-local dependencies of mesh vertices. This autoregressive model can generate a 3D mesh with conditional contexts such as given object classes or images.

Recently, a part-aware textured mesh generation method called TM-Net has been proposed by Gao et al. [98]. This VAE-based model can generate a 3D textured mesh from a random sample in the latent space while having different textures compatible with shape parts. Note that mesh geometry and texture are decoupled into PartVAE and TextureVAE. The key part of TextureVAE comprises an encoder that maps inputs into two continuous feature maps and a decoder that uses the feature maps to reconstruct textures. Meanwhile, PartVAE, adopted from Gao et al. [130] is used for encoding the detailed part geometry and resembles SP-VAE [130] in jointly encoding both global structure and part geometries.

3.3.4 *Summary*

Meshes are the most popular 3D shape representation in computer graphics, with the advantage of having additional topological information implied by mesh connectivity. However, this is also the main drawback in deep-learning-based mesh generation tasks. How to overcome the mesh topology constraint while not causing geometric and topological errors during the shape generation process is the key problem to solve. In contrast, other shape representations such as voxels, point clouds, and implicit functions, are more flexible in terms of shape topology, allowing more freedom for shape generation.

3.4 Implicit-Representation-based Shape Generation

3.4.1 *Basics*

Shape generation based on implicit representations has also attracted attention in recent years. While implicit representations cannot explicitly exhibit the underlying shape, the Marching Cubes algorithm [152] provides a general way to convert an implicit representation to an explicit representation. Meanwhile, the advantage of representing accurate geometry and flexible topology largely benefits work on 3D shape generation.

3.4.2 *Encoder-Decoder*

The two most commonly used implicit representations are the occupancy function and the signed distance function (SDF). The Occupancy Network (OccNet) [103] uses the occupancy function, having value 1 inside the surface and 0 outside. Similarly, DeepSDF [102] and DISN [104] represented a 3D shape by the signed distance to the underlying surface, dividing space into three regions: inside (SDF < 0), outside (SDF > 0), and on the surface (SDF $= 0$). Other implicit representations are based on level sets [201], the unsigned distance function [202], closest surface-point (CSP) [203], and probabilistic directed distance fields [204]. A neural-network-based generator is well suited by occupancy and SDF as they are both continuous functions defined within a 3D regular domain. As a result, implicit representations have soon been exploited to learn and generate 3D shapes using classic encoder-decoder architecture [104–111, 113].

IM-Net [113] learns the implicit field by a simple yet effective decoder composed of MLPs that returns the value showing the status (inside or outside) of the query point. Note that such a decoder can be used in AE, VAE, or GAN as the shape generator. Concurrent work, DISN [104], proposed a deep learning network to learn an implicit function for generating a high-quality 3D shape from a single image. It uses the local feature of projected point location, a global feature, and point features from MLPs to produce the SDF. Liu et al. [105] proposed a method that generates 3D shapes using implicit functions, without 3D supervision. The input image is first encoded into a latent vector by ResNet18. Then an implicit decoder consisting of 6 full-connected layers correlates the latent vector with a 3D query point and infers the occupancy probability. Later, Peng et al. [106] involved volume convolution to improve OccNet [103], while moving least-squares (MLS) functions are adopted in IMLSNet [107] for generating shapes from noisy inputs. Unlike previous works [102, 103, 113], IF-Net [108] uses Euclidean-space-aligned deep features and classifies deep features at continuous

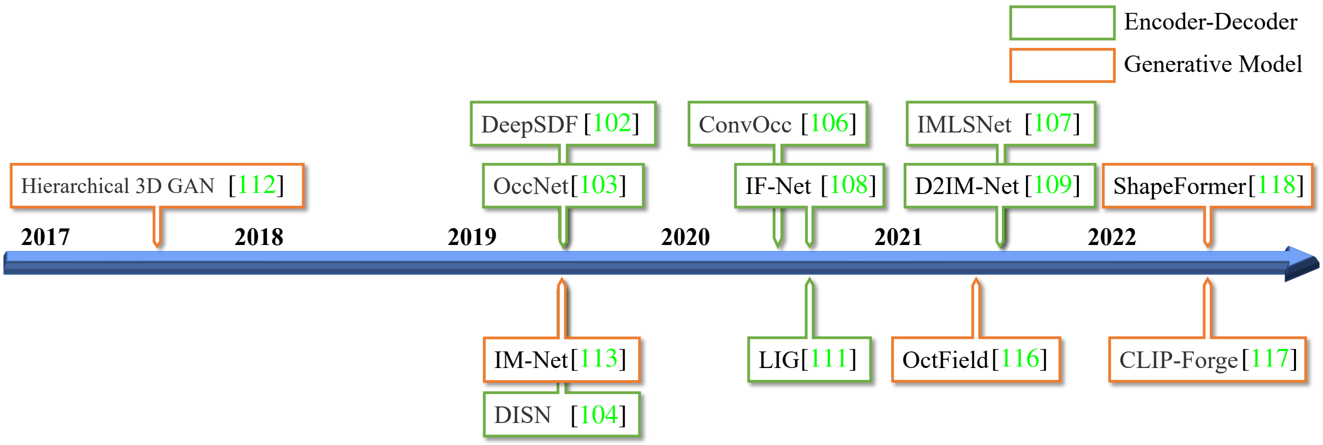
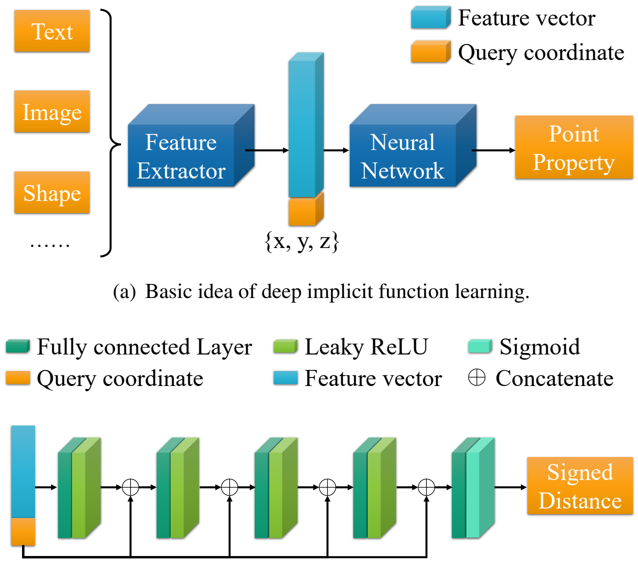


Fig. 15 Timeline of shape generation methods based on implicit representation.



(a) Basic idea of deep implicit function learning.

(b) Network structure of decoder in IM-Net [113].

Fig. 16 (a): Basic idea of deep implicit function learning. A query coordinate with relevant prior is fed into the network, which produces a point property, typically a single value. The network in [103] gives occupancy; the one in [102] is based on signed distance function. (b): The implicit function decoder in [113]. Simple MLPs using concatenated prior knowledge and query point coordinates as input can generate a high-quality inside-outside field.

query points.

As an extension of IM-Net, Li et al. [109] presented a pipeline, D2IM-Net, for shape generation from a 2D image based on an implicit function that can decouple details and global features. A CNN-based encoder extracts global and local features from the input image. While global features are fed into a decoder with the query point to generate a coarse shape as an SDF, another decoder for local features produces two displacement maps (back and front) used to generate details on the coarse shape.

Poursaeed et al. [110] proposed a shape generation method that combines explicit and implicit representations and generates two shape representations simultaneously. After the input is fed into different encoders, the encoded features are sent to OccNet [103] and AtlasNet [83] branches to generate results. Consistency loss is adopted here to couple two different networks together. Experiments show that the hybrid model works better than the individual branches.

Differing from the aforementioned works, a local implicit grid (LIG) representation was proposed by Jiang et al. [111], which extends implicit shape generation to whole scenes. Benefiting from implicit representation, an implicit decoder trained with a 3DCNN encoder under a local grid structure can be generalized to large scenes and unseen objects. Qualitative results show that the local implicit grid can generate higher-quality shapes than a trivial implicit representation. This work also pioneered reconstructing scenes from sparse point sets in a scalable manner.

3.4.3 Generative Models

A GAN model was utilized in the early work of Jiang et al. [112] to generate shapes with details from an implicit representation. A hierarchical architecture was proposed to generate a coarse signed distance field and high-frequency details separately. Specifically, the low-frequency generator takes an up-convolution neural network as the backbone. It then passes the filtered generated results to the high-frequency generator as a condition to produce shapes with details.

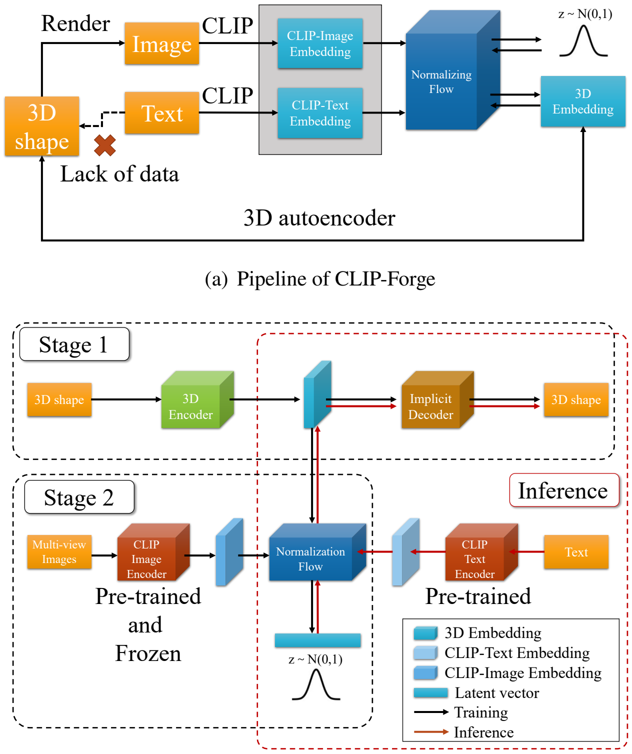
While IM-NET [113] can be employed in an autoencoder architecture, it can also be adapted to GAN-based models. Achlioptas et al. [57] train a latent-GAN on high-dimension shape features with a pre-trained autoencoder serving for dimension reduction. In addition, normal Wasserstein GAN loss is applied for the generative model in IM-Net. Resultant meshes extracted by Marching Cubes [152] show good mesh quality and flexible mesh topology (see Fig. 16). In other work [114] based on IM-Net, in the autoencoder structure, the encoder receives a high-resolution binary voxel and maps it into a grid of latent vectors. The decoder evaluates the implicit function represented by this grid on query points.

Mezghanni et al. [115] proposed a physically-aware generative network for shape generation. Novel physical losses are used to enhance the physical validity of generated shapes. Moreover, a joint latent space coding of geometry, structure, and physics is built for physically-aware generation. A physics module consisting of a topology layer and neural stability predictor is used for loss computation. Following the design of latent-GAN [57], the generator and discriminator consist of three fully-connected layers with WGAN-GP loss. The trained decoder uses random samples from latent space to decode 3D shapes.

As well as GAN-based models, autoregressive models are popular in shape generation. Yan et al. [118] proposed ShapeFormer, a transformer-based architecture to model conditional distribution by a sequence consisting of quantized features from the encoder. Furthermore, an implicit function called the vector quantized deep implicit function (VQDIF) was introduced, which can compress shape into a sequence of sparse local features. After autoregressive sampling, the VQDIF decoder converts this sequence back to the deep implicit function used for mesh extraction.

VAE-type methods are also used in implicit shape generation. The OctField [116] adopted the octree data structure to improve generation results. Both the encoder and decoder employ a hierarchical structure. During encoding, three layers consisting of MLP and max-pooling build a bridge for transferring information between child and parent nodes in the octree. During decoding, the feature from the parent octant is decoded into features with two indicators used for inferring the probability of surface occupancy and the necessity of the following subdivision through two MLPs with classifiers. Compared to the local implicit approach of Jiang et al. [111], OctField uses less memory and provides better modeling accuracy.

Taking advantage of CLIP [205] in cross-modality with deep implicit functions, Sanghi et al. [117] proposed a complete pipeline, CLIP-Forge, to generate shapes from text (see Fig. 17). It consists of three parts. First, an autoencoder composed of a voxel encoder and an implicit decoder is trained. Second, a pre-trained CLIP image encoder is used to train a conditional normalizing flow model with latent information from images and the above autoencoder. In the last, inferencing, part, the text input is coded by a pre-trained CLIP text encoder and fed into the flow model trained in the second part. The reversibility of the flow model makes it easy to produce corresponding latent information, which can be decoded into a 3D shape by a decoder trained in the first part. Note that the encoder and decoder are exchangeable to suit different output representations, like a point cloud.



(a) Pipeline of CLIP-Forge

(b) TNetwork architecture of CLIP-Forge

Fig. 17 CLIP-Forge [117] uses CLIP space to bridge text, image, and 3D shapes, to overcome the lack of paired text and shape training data. In (b), stage 1 is the training procedure for the 3D autoencoder, and stage 2 shows how CLIP is used for normalization flow model training.

Not using the standard generative model, Liu et al. [119] proposed a 3D shape generation method with text guidance using implicit maximum likelihood estimation (IMLE). Specifically, the network generates 3D shapes in occupancy representation with colors following the text description. It first adopts a shape autoencoder from IM-Net to extract shape and color features. Then, these two features are sent into a text-guided module with a shape and color decoder and a word-level spatial transformer (WLST). Local features from the WLST help improve the spatial correlation implied by the input text. For generation, a style-based latent shape generator accepts features from encoders and generates diverse 3D shapes.

3.4.4 Summary

Due to their continuous and regular definition, implicit representations have good compatibility with deep learning. In the task of 3D shape generation, we can also perform continuous and smooth interpolation in the hidden space. It can be found that training an encoder-decoder structure based on implicit representation often results in a smooth and continuous latent space. Although the implicit representation is not intuitive (cannot be explicitly visualized and modified), and requires additional work to convert it into an explicit representation, 3D shapes generated using implicit representation are still flexible and of relatively high quality.

3.5 Structure-based Shape Generation

3.5.1 Basics

Structure-based models have recently achieved higher performance and quality in 3D shape generation due to their unique features. Shape primitives are commonly used in structure-based representation by describing a 3D shape using simple primitives like oriented bounding boxes. Even though details are less of concern in this representation, more attention is paid to the global shape structure.

3.5.2 Encoder-Decoder

Researchers have attempted to encode 3D shape structures and geometric features individually or jointly. In early work, 3D-PRNN, Zou et al. [120] came up with the idea of generating and combining 3D object parts in primitives using generative recurrent neural networks. First, they encode the depth image of the target object. They then feed the latent code to several mixture density network modules with LSTM to sequentially generate a primitive set. Eventually, they combine the primitives to build the 3D shape into the target.

CompoNet [121] generates point clouds in a structure-based way. More specifically, they use parallel generative autoencoders to learn part synthesis information first. After that, a noise vector is fed into the part composition network and then concatenated with the latent output from the part synthesis unit. Finally, a fully-connected layer is used to generate points (400×3 dimensions) and all parts are warped to generate the sample shape.

Unlike another sequence based method 3D-PRNN [120], PQ-NET [122] learns both structure combination and geometries of individual parts. To generate geometric shapes, while a CNN-based encoder is employed for input image coding, a decoder composed of an MLP generates implicit functions which can be sampled at different resolutions. A bidirectional stacked RNN [206] with GRU [207] is used to build the encoder and decoder, which learns to assemble and decompose in both ways for the autoencoder. Although PQ-NET can generate shapes while decoupling structure and geometry, it cannot learn global relations like symmetry and cannot change topology during shape interpolation.

Unlike previous structure-based methods [121, 122], COALESCE by Yin et al. [123] aligns parts and jointly synthesizes an implicit surface for 3D shape assembly. With an IM-Net decoder predicting an implicit surface, COALESCE focuses on learning to generate output seamlessly with the given point cloud parts.

The Neural Star Domain (NSD) [124] can be regarded as using continuous functions defined on the surface of a sphere. As a unified shape representation, NSD can define both implicit and explicit shape representations of a primitive shape. The NSD Network (NSDN) was proposed to generate structured shapes within the neural star domain. Like OccNet, a bottleneck auto-encoder is employed in NSDN to map images to shape embedding, and a translation network outputs translation vectors from the shape embedding, then the translation vector along with the embedded feature and the given angular coordinates are used for pose inference of the primitives. As a result, they achieved state-of-the-art single-view reconstruction results.

3.5.3 Generative Models

In early work, Kalogerakis et al. [133] used a probabilistic model to synthesize complex shapes. The model can represent shapes with structural variability within a particular domain. A new shape from this domain can be composed of existing components. Features are split into continuous and discrete features, and the model represents the joint probability distribution over random variables, which is factorized as a product of conditional probability distributions (CPDs). The training phase learns the model structure and the parameters of all CPDs. After training, synthesis of a set of components and optimization of component placement are applied to generate a new shape. Later, probabilistic-based methods like [134, 135] tried to learn from 3D shapes through part-based methods. By fitting a template, they estimate point correspondence, rigid alignment, segmentation, and shape a. In particular, Huang et al. [135] constructed the template with newly learned parts equipped with probabilities while decoupling shape structure and geometry.

Differing from such probabilistic-based works [133–135], Sung et al. [136] proposed ComplementMe, another strategy to automatically synthesize shapes from the input. They apply two networks for shape retrieval and placement to perform incremental shape assembly. While the retrieval

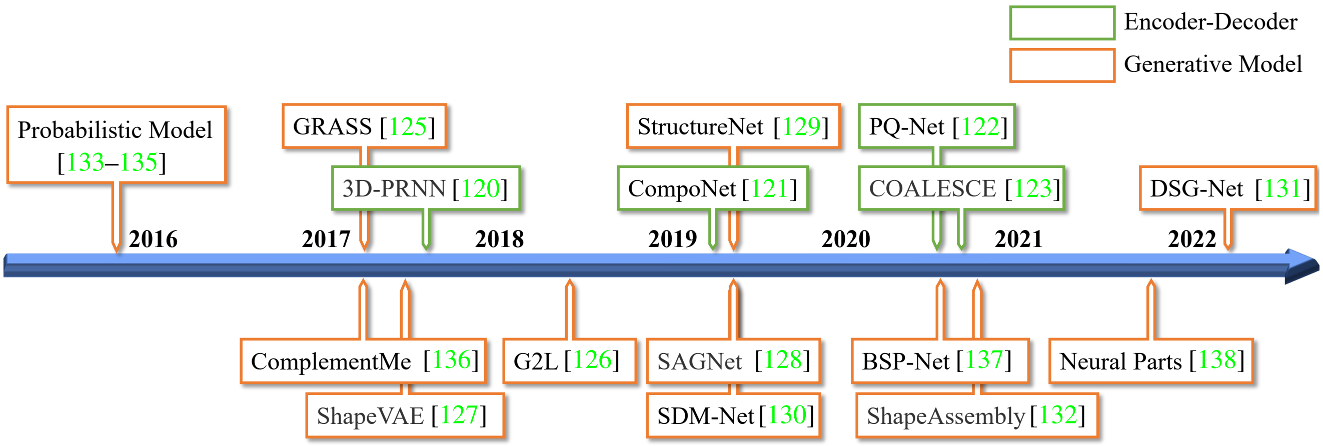
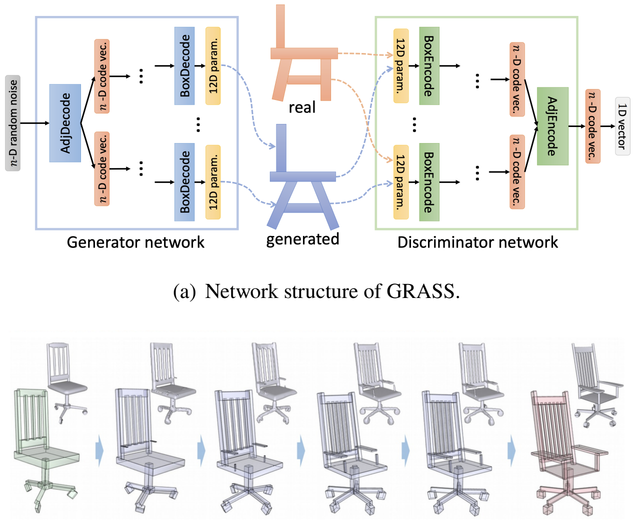


Fig. 18 Timeline of shape generation methods based on structural representation.



(a) Network structure of GRASS.

(b) Structure interpolation.

Fig. 19 The Generative Recursive Autoencoder for generating Shape Structures (GRASS) [125] can interpolate two shapes by operating on the fixed-length codes from the autoencoder. Images courtesy of [125].

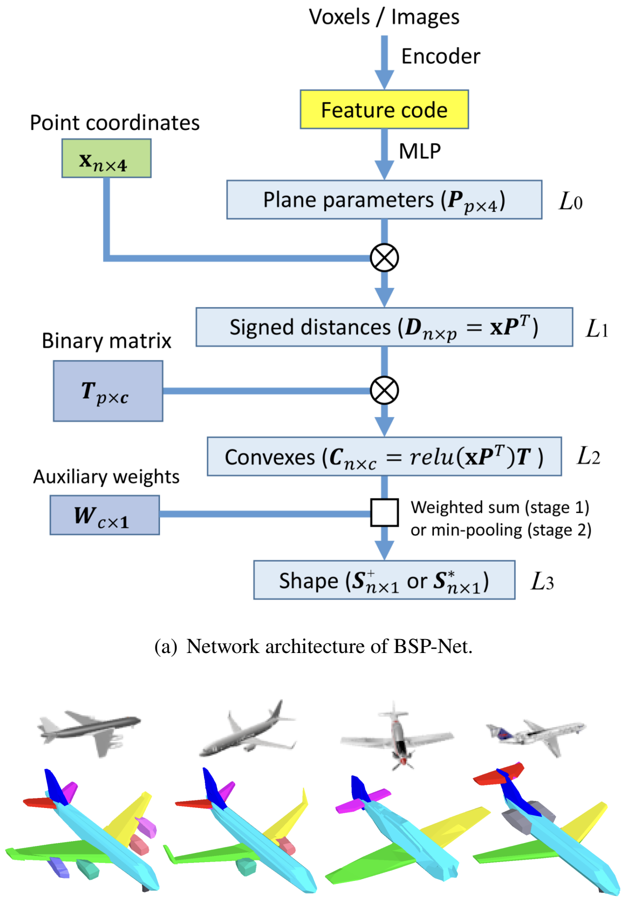
network predicts the probability distribution for the input partial shape and samples the component complement from the distribution, the placement network puts all components in the right position for a complete shape. PointNet is used for both retrieval and placement networks.

Nash et al. [127] proposed a variational autoencoder called shapeVAE for 3D shape related tasks such as completion and alignment. Since then more structure-based deep generative models have been explored. GRASS (Generative Recursive Autoencoders for Shape Structures) [125] was proposed to describe full shapes by a binary tree. This work was one of the very first to encode shapes by neural networks. A pre-trained RvNN autoencoder is used to obtain root codes in the tree, and a GAN module (see Fig. 19) is applied to learn the manifold with the code space. Finally, a volumetric network generates detailed geometry from the synthesized OBBs even though the binary tree can overflow easily when the data size grows.

To consider geometry and structure of shape simultaneously, Wu et al. [128] proposed SAGNet, a generative model jointly learning geometry and pairwise relationships of shape parts. Specifically, the input 3D shape is sent into a 3D convolutional and fully-connected layer to extract features. Then, the GRU-based encoder with an attention component jointly learns the high-dimensional feature of geometry and structure. Finally, a 2-way VAE provides latent sampling for diverse generated results. Meanwhile, a hierarchical graph network is used to generate a unified latent space for shape encoding with structure and geometry variations as in StructureNet [129].

Wang et al. [126] employed a global-to-local(G2L) approach for 3D voxel-based shape generation. A GAN module generates a 32 × 32 × 32 volume representing a global shape. At the same time, global and local discriminators score both the whole shape and the individual parts under quality losses. Afterwards, the generated shape is fed into a part refiner (PR) consisting of an encoder and decoder composed of convolution and fully connected layers to complete the shape and increase the resolution to 64 × 64 × 64.

A BSP-tree-based method [137] was proposed later concerning convexes pieces of shape geometry (see Fig. 20). Convex pieces as a new form of primitives involve three layers: hyperplane extraction, hyperplane grouping, and shape assembly, and can better represent 3D shape details than prior work like StructureNet [129]. Another stream of work aims to decouple geometry and structure of shapes. SDM-Net [130] and DSG-Net [131] both use VAEs as their generator to limit the interpolation space and provide outstanding results in


Voxels / Images
Encoder
Feature code
Point coordinates
$\mathbf{x}_{n\times4}$
MLP
Plane parameters ($\boldsymbol{P}_{p\times4}$)
L_0
Signed distances ($\boldsymbol{D}_{n\times p} = \mathbf{x}\boldsymbol{P}^T$)
L_1
Binary matrix
$\boldsymbol{T}_{p\times c}$
Convexes ($\boldsymbol{C}_{n\times c} = relu(\mathbf{x}\boldsymbol{P}^T)\boldsymbol{T}$)
L_2
Auxiliary weights
$\boldsymbol{W}_{c\times1}$
Weighted sum (stage 1)
or min-pooling (stage 2)
Shape ($\boldsymbol{S}^+_{n\times1}$ or $\boldsymbol{S}^*_{n\times1}$)
L_3

(a) Network architecture of BSP-Net.

(b) 3D shape generation from a single image.

Fig. 20 Structured single view reconstruction based on BSP-Net [137]. Convex pieces having the same color share the same shape semantics. Images courtesy of [137].

geometry and structure generation and decoupling.

Later works can generate high-quality structure and geometry while controlling them to some extent. Yang et al. [131] proposed DSG-NET to decompose a shape into two different decoupled latent spaces based on the Cycled Disentanglement mechanism. It can synthesize shapes while controlling structure and geometry (see Fig. 21). While the network design is inspired by SDM-Net [130], the decoder reconstructs part geometry by decoding the ACAP feature [208] and the center vector. Note that both encoder and generator are trained in a decoupled but synergistic manner.

In a different approach to previous structure-based methods, Jones et al. [132] proposed ShapeAssembly which defines a domain-specific language for shapes. In the ShapeAssembly program, a hierarchical sequence VAE is adopted. It consists of a decoder including multiple line decoders based on RNNs. Moreover, as a language-like system, a differentiable interpreter for ShapeAssembly is provided to help generate the desired shape parts.

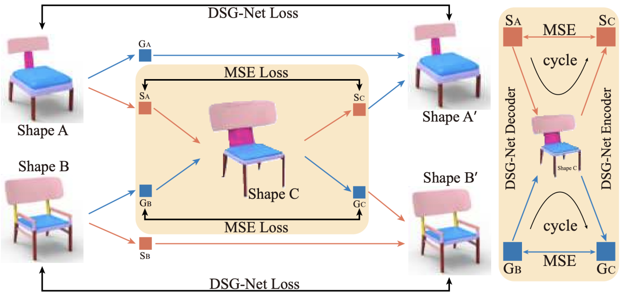


Fig. 21 DSG-Net decouples shapes into geometry and structure features. New shapes can be synthesized by fusing decoupled geometry and structure features. Image courtesy of [131].

The structure-based approach can also be used for human parts generation. Paschalidou [138] proposed Neural Parts that can generate 3D shapes with invertible neural networks. Given an input image, the proposed network can express the target object with several primitive shapes deformed from a sphere. The first feature extractor maps the input to a global feature representation and combines the global feature with learnable embedding to generate the shape embedding for each primitive. Then, a conditional homeomorphism component consisting of several conditional coupled layers generates points on the surfaces. An inverse mapping operator can help compute point positions in 3D space related to the primitive. Such inverse mapping can involve additional constraints on the predicted objects. High-quality reconstructions are experimentally demonstrated, showing the increased effectiveness compared to other methods.

3.5.4 Summary

Structure-based methods usually generate 3D shapes with good quality due to modeling the detailed shape structure using primitives. However, the cost is also considerable. First, they usually require additional annotations on shape segmentation, making it sensitive to shape noise within a dataset, and shape variations between datasets. Moreover, some methods use generative models to generate parts before composing the final shape, and often suffer from low quality of the generated parts. Although some methods exploit shape deformation at a fine-grained level, they can only handle global topology during the assembly process but not at the primitive level. Therefore, the generated shape can only meet global structural requirements, but lack detailed control of part geometry. Lastly, the number of shape primitives they can handle is often restricted by the limited parameters of the network, which affects the complexity of the generated shapes.

4 Datasets for 3D Generation

In addition to 3D geometry learning techniques, researchers have also established a variety of datasets that meet the need for training deep learning models. Using an appropriate dataset can help the shape learning and generation process according to the application.

There are several critical challenges for building 3D shape training datasets for deep learning models. First, compared to easy-to-collect, low-cost 2D image datasets, obtaining corresponding 3D data is difficult. Many 2D datasets cannot be directly used for 3D generation tasks (due to the emergence of differentiable rendering [113, 198, 199], although some 2D image datasets [220] can also be used for 3D deep learning after specific preprocessing, but are not the main focus of the present survey). The most commonly used datasets, such as ShapeNet [2], ModelNet [13], and PASCAL 3D+ [215] usually contain 3D CAD models. It is often difficult to obtain corresponding real-world images. For supervised image-based 3D shape generation tasks, in the absence of such natural images, only synthetic images made by projecting and rendering 3D models can be obtained. Finally, there is no dataset that can meet all requirements for various shape representations at the same time. The shapes in the dataset need to be preprocessed (e.g. voxelized, sampled) in order to be used for different representations.

Here we summarise widely-used datasets for deep 3D shape generation; we also include self-contained image data which can be used for image-based 3D shape generation. A summary of dataset statistics can be found in Table 2.

ShapeNet [2] is the most commonly used dataset for geometry learning in recent years. It consists of 3,000,000 computer-aided design (CAD) models; 220,000 models have been organized into 3135 categories, where 3D models and 2D images are aligned. In addition, there are also smaller subsets of ShapeNet for research use, such as ShapeNetCore (55 object categories, 51,300 models with verified category and alignment annotations) and ShapeNetSem (270 categories, 12,000 models with verified category and consistent alignments including real-world dimensions, category-level material composition estimation, and volume/weight estimation).

PartNet [213] is usually used for point cloud related tasks. PartNet has a total of 16 classes, 50 parts, and a total of 16,846 samples.

ModelNet [13] is another well-known extensive shape dataset of 3D CAD models. As the most widely used benchmark for point cloud analysis, ModelNet40 is popular because of its wide range of categories, high quality shapes, etc. The point cloud data are uniformly sampled from corresponding mesh surfaces and then further preprocessed by moving to the origin and scaling to a unit sphere.

ObjectNet3D [210] is a large scale dataset of 3D shapes and 2D images. 2D images are aligned with the corresponding 3D shapes. The alignment provides both accurate 3D pose annotation and the closest 3D shape annotation for each 2D image.

TOSCA [216], a high-quality non-rigid 3D mesh shape datasets, contains 80 models in 9 categories. The typical number of mesh vertices is about 50,000. Models within the same class have compatible triangulations with the same number of vertices. This feature can also be used for shape matching with point correspondences.

FAUST [217] is a human body dataset. It contains 300 real human scans of 10 different subjects in 30 different poses, acquired with a high-accuracy 3D multi-stereo system. While having different identities and aligned with various poses, these real-world scans are noisy and incomplete.

SUNCG [211] is dataset of 3D indoor scenes. It is a manually created large-scale dataset of 3D synthetic scenes with dense volumetric annotations that can be used for semantic segmentation, depth estimation, visual navigation, etc. It contains 2644 unique scenes with 5,697,217 object instances.

3D Future [214] is a large-scale furniture dataset that contains 20,240 synthetic images captured from 5,000 diverse scenes, and 9,992 unique 3D industrial furniture shapes with high-resolution textures. It also provides instance segmentation annotation and image rendering information, including intrinsic and extrinsic camera parameters.

Pix3D [212] is a dataset for single-image-based 3D shape modeling which provides precise 2D to 3D alignment. There are 395 3D shapes in 9 categories with 10,069 image-shape pairs with precise 3D annotation.

SUN RGB-D [218] is a dataset that includes 10,335 RGB-D images with both 2D and 3D annotations for scene categorization, semantic segmentation, and other popular tasks. It contains 47 scene categories and 800 object categories.

A Large Dataset of Object Scans [219] is a multi-modal 3D dataset captured by consumer-level mobile 3D scanning setups. It has a total of 10,933 RGB-D scans, 398 reconstructed models, and 10,933 videos created from images.

5 Discussion

After the review of existing work on deep-learning-based 3D shape generation, in this section, we discuss several potential directions that hopefully can inspire future work in this area.

Table 2 Datasets for deep 3D shape generation.

Dataset	Year	3D Data Type	#Object Category	Size	Source	Description
ShapeNet [2]	2015	3D CAD Models	3,135	3,000,000	Synthesis	Large scale dataset, rich annotations.
ShapeNetCore [2]	2015	3D CAD Models	55	51,300	Synthesis	Subset of ShapeNet with clean models and alignment annotations.
ShapeNetSem [2]	2015	3D CAD Models	270	12,000	Synthesis	Subset of ShapeNet annotated with real-world dimensions (volume, weight, etc.)
ModelNet [13]	2015	3D CAD Models	662	127,915	Synthesis	Manual classification
ModelNet10 [13]	2015	3D CAD Models	10	4,899	Synthesis	Orientation aligned, completely cleaned
ModelNet40 [13]	2015	3D CAD Models	40	12,311	Synthesis	Orientation aligned version produced by [209]
ObjectNet3D [210]	2016	3D CAD Models	100	44,147	Synthesis	2D images are aligned with 3D objects.
SUNCG [211]	2017	3D CAD Models	84	5,697,217	Synthesis	Dataset of large-scale scenes, dense volumetric annotations.
Pix3D [212]	2018	3D CAD Models	9	395	Synthesis	2D images are aligned with 3D objects
PartNet [213]	2019	3D CAD Models	24	26,671	Synthesis	ShapeNet with fine-grained, hierarchical instance-level 3D part annotations.
3D-FUTURE [214]	2020	3D CAD Models	-	9,992	Synthesis	Furniture with high-resolution textures. 2D images are aligned with 3D objects.
PASCAL3D+ [215]	2014	3D CAD Models	12	36,000	Synthesis	Models annotated with dense pose and occlusion-aware information.
TOSCA [216]	2008	Non-rigid Models	9	80	Synthesis	Models in the same class have per-vertex correspondence.
FAUST [217]	2014	Non-rigid Models	10	300	Real scans	Human bodies
SUN RGB-D [218]	2015	RGB-D images	800	10,335	Real scans	RGB-D image with rich annotations and bounding boxes.
Object Scans [219]	2016	RGB-D images	44	23,000,000	Real scans	Dataset includes images, reconstructed meshes, and videos.

5.1 Multi-Representational Learning

Existing representations for learning 3D shape generation include voxels, point clouds, meshes, implicit functions, and structure-based representations. Nevertheless, each representation has its own limitations, which affect geometric details or shape structure, or limit the design of the overall generative network. Although the flexibility of deep implicit function learning [102, 103] facilitates shape generation to some extent, implicit functions can neither explicitly express nor allow intuitive editing of output surfaces as easily as meshes. On the other hand, mesh representation also has its own disadvantage of usually requiring a fixed topology in works based on mesh template deformation [85, 86]. To break such a bottleneck, researchers have started to combine different representations. Shen et al. [221] first use SDF to predict the initial surface and then refine it by graph convolution networks. A surface loss is applied to train the explicit surface from the differentiable marching tetrahedra layer to achieve refined surfaces. Similarly, to overcome the drawback of implicit fields, Yuan et al. [222] achieve explicit surface editing by manipulating a NERF architecture [11], resulting in well-controlled editing and rendering results. In other work, Hui et al. [91] leverage the strengths of both mesh-based and structure-based methods. Given meshes are easy to deform yet preserve fine details, but it is hard to change their topology, Structure-based representations can help to identify the closest shapes having the correct topology as initialization for mesh deformation, which is critical for mesh-deformation-based methods. To sum up, exploiting the advantages while compensating for the weaknesses of different representations would be a valuable direction to explore.

5.2 Higher Quality 3D Shape Generation

Existing works, albeit generating 3D shapes with good characteristics, still lack the ability to generate geometric details. Generative networks based on voxels and implicit representations require massive computing resources to generate high-quality shapes with details. The Marching Cubes method [152, 223] also has certain limitations in the embodiment of details. Compared to voxel and implicit representations, point clouds relies on a large number of 3D points to represent a complete shape. However, the shape uncertainty caused by the lack of local topological connections and the ambiguous overall topology of the point cloud largely affects the accuracy of the underlying 3D shapes. Although meshes balance shape geometry and topology, its irregularity poses challenges in designing generative networks and heavily influences details of the generated shapes. Structure-based representation controls the overall structure and global features of 3D shapes but is limited by the network characteristics and result assembly [125, 131]. Hence it cannot sufficiently control details of the generated shapes. Based on the above observations, we believe there is still a need for more research on how to better control the quality of generated shapes and how to enable the generator to learn more precise details while capturing overall shape features.

5.3 Larger 3D Scene Generation

While most generators aim to generate single object, a few 3D generative networks attempt to generate large scenes with multiple objects. The main challenge is that the data volume needed for a 3D scene is usually huge., and such a large amount of data is infeasible as the output of a generative network. Some scene generation networks use prior knowledge such as scene semantics and object orientations to match objects in the dataset to specific locations to form the generated scene [224], but it is challenging to control the overall quality and object compatibility. Existing works like [60, 225] using point clouds and implicit fields are the most likely to generate scene-level data in this regard. Point clouds and implicit fields can be up-sampled to improve the representation quality while not grasping the relationship between scene objects. Also, structure-based shape representation has the potential to be extended and learned at the scene level to express spatial relations between scene objects and shape characteristics of individual objects.

5.4 3D Backbone Network Design

For the generation of 3D shapes, a good backbone network should be able to simultaneously encode 3D shapes into latent embeddings and, at the same time, recover better shapes from latent embeddings. Proposal of new backbone networks plays an important role in generating 3D shapes. Every revolution in 3D model generation occurs after a novel, efficient backbone network is proposed. Most generators using the same type of representation share similar output layers, such as voxel-based works [15, 16, 28, 32], point-cloud-based works [49, 147, 176], MeshCNN [149] and SubdivNet [151] for mesh representation learning, and OccNet [103] and DeepSDF [102, 104] for implicit representations. Very recently, the transformer mechanism has also been employed in 3D deep learning [141, 226] with demonstrated advantages compared to classic 3D convolutional networks. The backbone network architecture usually determines the shape representation, which in turn affects the design of the generator. 3D shape generation benefited from the development of backbone networks for different shape representations and will also inspire exploration of more generalized and effective 3D backbone network design.

6 Conclusions

This survey has given a detailed survey of the development of deep-learning-based shape generation. First, we outlined several commonly used 3D representations and popular deep learning models for data generation. Next, we reviewed existing works according to the shape representation and the generator used for generating 3D shapes. We have discussed properties of the shape representation, the architecture of the shape generator, and the characteristics of the results. Benefits and limitations have also been analyzed. We have also covered widely used public datasets for 3D shape generation. Lastly, we have suggested a few future research directions. We hope this survey has given interested readers a brief overview of the field and provides inspiration for future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No.: 61902210) and RCUK grant CAMERA (EP/M023281/1, EP/T022523/1).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Zhang Z. Microsoft kinect sensor and its effect. *IEEE Multimedia*, 2012, 19(2): 4–10.

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

[3] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, 248–255.

[4] Kirk D, et al.. NVIDIA CUDA software and GPU parallel computing architecture. In *Proceedings of the International Symposium on Memory Management (ISMM)*, volume 7, 2007, 103–104.

[5] Guo MH, Xu TX, Liu JJ, Liu ZN, Jiang PT, Mu TJ, Zhang SH, Martin RR, Cheng MM, Hu SM. Attention mechanisms in computer vision: A survey. *Computational Visual Media*, 2022, 8(3): 331–368.

[6] Cao W, Yan Z, He Z, He Z. A comprehensive survey on geometric deep learning. *IEEE Access*, 2020, 8: 35929–35949.

[7] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 4401–4410.

[8] Ramesh A, Pavlov M, Goh G, Gray S, Voss C, Radford A, Chen M, Sutskever I. Zero-shot text-to-image generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021, 8821–8831.

[9] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[10] Huang J, Huang SS, Song H, Hu SM. Di-fusion: Online implicit 3d reconstruction with deep priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 8932–8941.

[11] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, 405–421.

[12] Saito S, Huang Z, Natsume R, Morishima S, Kanazawa A, Li H. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, 2304–2314.

[13] Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, 1912–1920.

[14] Yan X, Yang J, Yumer E, Guo Y, Lee H. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Proceedings of the 29th International Conference on Neural Information Processing Systems (NeurIPS)*, 2016, 1696–1704.

[15] Girdhar R, Fouhey DF, Rodriguez M, Gupta A. Learning a predictable and generative vector representation for objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, 484–499.

[16] Choy CB, Xu D, Gwak J, Chen K, Savarese S. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, 628–644.

[17] Wu J, Wang Y, Xue T, Sun X, Freeman B, Tenenbaum J. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS)*, 2017, 540–550.

[18] Wu J, Zhang C, Zhang X, Zhang Z, Freeman WT, Tenenbaum JB. Learning shape priors for single-view 3d completion and reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, 646–662.

[19] Zhang X, Zhang Z, Zhang C, Tenenbaum J, Freeman B, Wu J. Learning to reconstruct shapes from unseen classes. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2018, 2263–2274.

[20] Kar A, Häne C, Malik J. Learning a multi-view stereo machine. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS)*, 2017, 365–376.

[21] Liu S, Giles L, Ororbia A. Learning a hierarchical latent-variable model of 3d shapes. In *Proceedings of 2018 International Conference on 3D Vision (3DV)*, 2018, 542–551.

[22] Tatarchenko M, Dosovitskiy A, Brox T. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, 2088–2096.

[23] Häne C, Tulsiani S, Malik J. Hierarchical surface prediction for 3d object reconstruction. In *Proceedings of 2017 International Conference on 3D Vision (3DV)*, 2017, 412–420.

[24] Cao YP, Liu ZN, Kuang ZF, Kobbelt L, Hu SM. Learning to reconstruct high-quality 3D shapes with cascaded fully convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, 616–633.

[25] Liu ZN, Cao YP, Kuang ZF, Kobbelt L, Hu SM. High-quality textured 3D shape reconstruction with cascaded fully convolutional networks. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2019, 27(1): 83–97.

[26] 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 International Conference on Computer Vision (ICCV)*, 2019, 2690–2698.

[27] Yang S, Xu M, Xie H, Perry S, Xia J. Single-view 3D object reconstruction from shape priors in memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 3152–3161.

[28] Wu J, Zhang C, Xue T, Freeman B, Tenenbaum J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Proceedings of the 29th International Conference on Neural Information Processing Systems (NeurIPS)*, 2016, 82–90.

[29] Smith EJ, Meger D. Improved adversarial systems for 3d object generation and reconstruction. In *Proceedings of the Conference on Robot Learning*, 2017, 87–96.

[30] Zhu JY, Zhang Z, Zhang C, Wu J, Torralba A, Tenenbaum J, Freeman B. Visual object networks: Image generation with disentangled 3D representations. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2018, 118–129.

[31] Chen K, Choy CB, Savva M, Chang AX, Funkhouser T,

Savarese S. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, 2018, 100–116.

[32] Knyaz VA, Kniaz VV, Remondino F. Image-to-voxel model translation with conditional adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, 601–618.

[33] Gadelha M, Maji S, Wang R. 3d shape induction from 2d views of multiple objects. In *Proceedings of 2017 International Conference on 3D Vision (3DV)*, 2017, 402–411.

[34] Li X, Dong Y, Peers P, Tong X. Synthesizing 3d shapes from silhouette image collections using multi-projection generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 5535–5544.

[35] Khan SH, Guo Y, Hayat M, Barnes N. Unsupervised primitive discovery for improved 3D generative modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 9739–9748.

[36] Henzler P, Mitra NJ, Ritschel T. Escaping plato's cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, 9984–9993.

[37] Chen Z, Kim VG, Fisher M, Aigerman N, Zhang H, Chaudhuri S. Decor-gan: 3d shape detailization by conditional refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 15740–15749.

[38] Brock A, Lim T, Ritchie JM, Weston N. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016.

[39] Balashova E, Singh V, Wang J, Teixeira B, Chen T, Funkhouser T. Structure-aware shape synthesis. In *Proceedings of 2018 International Conference on 3D Vision (3DV)*, 2018, 140–149.

[40] Mittal P, Cheng YC, Singh M, Tulsiani S. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, 306–315.

[41] Huang W, Lai B, Xu W, Tu Z. 3D volumetric modeling with introspective neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 2019, 8481–8488.

[42] Ibing M, Kobsik G, Kobbelt L. Octree Transformer: Autoregressive 3D Shape Generation on Hierarchically Structured Sequences. *arXiv preprint arXiv:2111.12480*, 2021.

[43] Xie J, Zheng Z, Gao R, Wang W, Zhu SC, Wu YN. Learning descriptor networks for 3d shape synthesis and analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, 8629–8638.

[44] Xie J, Zheng Z, Gao R, Wang W, Zhu SC, Wu YN. Generative VoxelNet: learning energy-based models for 3D shape synthesis and analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020, 44(5): 2468–2484.

[45] Gadelha M, Wang R, Maji S. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, 103–118.

[46] Yang Y, Feng C, Shen Y, Tian D. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, 206–215.

[47] Zamorski M, Zieba M, Klukowski P, Nowak R, Kurach K, Stokowiec W, Trzcinski T. Adversarial autoencoders for compact representations of 3D point clouds. *Computer Vision and Image Understanding*, 2020, 193: 102921.

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

[49] Fan H, Su H, Guibas LJ. 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 (CVPR)*, 2017, 605–613.

[50] Wei Y, Liu S, Zhao W, Lu J. Conditional single-view shape generation for multi-view stereo reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 9651–9660.

[51] Hu T, Lin G, Han Z, Zwicker M. Learning to generate dense point clouds with textures on multiple categories. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, 2170–2179.

[52] Lin CH, Kong C, Lucey S. Learning efficient point cloud generation for dense 3d object reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, 2018, 7114–7121.

[53] Insafutdinov E, Dosovitskiy A. Unsupervised learning of shape and pose with differentiable point clouds. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2018, 2807–2817.

[54] Chen C, Han Z, Liu YS, Zwicker M. Unsupervised learning of fine structure generation for 3d point clouds by 2d projections matching. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, 12466–12477.

[55] Komarichev A, Hua J, Zhong Z. Learning geometry-aware joint latent space for simultaneous multimodal shape generation. *Computer Aided Geometric Design*, 2022, 93: 102076.

[56] Gal R, Bermano A, Zhang H, Cohen-Or D. MRGAN: Multi-Rooted 3D Shape Generation with Unsupervised Part Disentanglement. *arXiv preprint arXiv:2007.12944*, 2020.

[57] Achlioptas P, Diamanti O, Mitliagkas I, Guibas L. Learning representations and generative models for 3d point clouds. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018, 40–49.

[58] Valsesia D, Fracastoro G, Magli E. Learning localized generative models for 3d point clouds via graph convolution. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[59] Shu DW, Park SW, Kwon J. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, 3859–3868.

[60] Li R, Li X, Fu CW, Cohen-Or D, Heng PA. Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, 7203–7212.

[61] Ramasinghe S, Khan S, Barnes N, Gould S. Spectral-gans for high-resolution 3d point-cloud generation. In *Proceedings of 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, 8169–8176.

[62] Li Y, Baciu G. SAPCGAN: Self-Attention based Generative Adversarial Network for Point Clouds. In *Proceedings of IEEE 19th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, 2020, 52–59.

[63] Li Y, Baciu G. Hsgan: Hierarchical graph learning for point cloud generation. *IEEE Transactions on Image Processing*, 2021, 30: 4540–4554.

[64] Li R, Li X, Hui KH, Fu CW. SP-GAN: Sphere-guided 3D shape generation and manipulation. *ACM Transactions on Graphics (TOG)*, 2021, 40(4): 1–12.

[65] Tang Y, Qian Y, Zhang Q, Zeng Y, Hou J, Zhe X. WarpingGAN: Warping Multiple Uniform Priors for Adversarial 3D Point Cloud Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, 6397–6405.

[66] Hui L, Xu R, Xie J, Qian J, Yang J. Progressive point cloud deconvolution generation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, 397–413.

[67] Arshad MS, Beksi WJ. A progressive conditional generative adversarial network for generating dense and colored 3D point clouds. In *Proceedings of 2020 International Conference on 3D Vision (3DV)*, 2020, 712–722.

[68] Wen C, Yu B, Tao D. Learning progressive point embeddings for 3d point cloud generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 10266–10275.

[69] Mo K, Wang H, Yan X, Guibas L. Pt2pc: Learning to generate 3d point cloud shapes from part tree conditions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, 683–701.

[70] Yang X, Wu Y, Zhang K, Jin C. Cpcgan: A controllable 3d point cloud generative adversarial network with semantic label generating. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, 2021, 3154–3162.

[71] Kim J, Yoo J, Lee J, Hong S. Setvae: Learning hierarchical composition for generative modeling of set-structured data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 15059–15068.

[72] Li S, Liu M, Walder C. EditVAE: Unsupervised Parts-Aware Controllable 3D Point Cloud Shape Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, 2022, 1386–1394.

[73] Yang G, Huang X, Hao Z, Liu MY, Belongie S, Hariharan B. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, 4541–4550.

[74] Klokov R, Boyer E, Verbeek J. Discrete point flow networks for efficient point cloud generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, 694–710.

[75] Kim H, Lee H, Kang WH, Lee JY, Kim NS. Softflow: Probabilistic framework for normalizing flow on manifolds. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*, 2020, 16388–16397.

[76] Pumarola A, Popov S, Moreno-Noguer F, Ferrari V. C-flow: Conditional generative flow models for images and 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, 7949–7958.

[77] Postels J, Liu M, Spezialetti R, Van Gool L, Tombari F. Go with the flows: Mixtures of normalizing flows for point cloud generation and reconstruction. In *Proceedings of 2021 International Conference on 3D Vision (3DV)*, 2021, 1249–1258.

[78] Sun Y, Wang Y, Liu Z, Siegel J, Sarma S. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, 61–70.

[79] Luo S, Hu W. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 2837–2845.

[80] Zhou L, Du Y, Wu J. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, 5826–5835.

[81] Xie J, Xu Y, Zheng Z, Zhu SC, Wu YN. Generative pointnet: Deep energy-based learning on unordered point sets for 3d generation, reconstruction and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 14976–14985.

[82] Cai R, Yang G, Averbuch-Elor H, Hao Z, Belongie S, Snavely N, Hariharan B. Learning gradient fields for shape generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, 364–381.

[83] Groueix T, Fisher M, Kim VG, Russell BC, Aubry M. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, 216–224.

[84] Pontes JK, Kong C, Sridharan S, Lucey S, Eriksson A, Fookes C. Image2mesh: A learning framework for single image 3d reconstruction. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, 2018, 365–381.

[85] Wang N, Zhang Y, Li Z, Fu Y, Liu W, Jiang YG. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, 52–67.

[86] Wen C, Zhang Y, Li Z, Fu Y. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, 1042–1051.

[87] Pan J, Han X, Chen W, Tang J, Jia K. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, 9964–9973.

[88] Shi Y, Ni B, Liu J, Rong D, Qian Y, Zhang W. Geometric Granularity Aware Pixel-To-Mesh. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, 13097–13106.

[89] Tang J, Han X, Pan J, Jia K, Tong X. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 4541–4550.

[90] Gkioxari G, Malik J, Johnson J. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, 9785–9795.

[91] Hui KH, Li R, Hu J, Fu CW. Neural Template: Topology-Aware Reconstruction and Disentangled Generation of 3D Meshes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, 18572–18582.

[92] Zhang SH, Guo YC, Gu QW. Sketch2Model: View-aware 3d modeling from single free-hand sketches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 6012–6021.

[93] Chen W, Ling H, Gao J, Smith E, Lehtinen J, Jacobson A, Fidler S. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, 2019, 9605–9616.

[94] Grigorev A, Iskakov K, Ianina A, Bashirov R, Zakharkin I, Vakhitov A, Lempitsky V. Stylepeople: A generative model of fullbody human avatars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 5151–5160.

[95] Pavllo D, Spinks G, Hofmann T, Moens MF, Lucchi A. Convolutional generation of textured 3d meshes. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*, 2020, 870–882.

[96] Pavllo D, Kohler J, Hofmann T, Lucchi A. Learning Generative Models of Textured 3D Meshes from Real-World Images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, 13879–13889.

[97] Tan Q, Gao L, Lai YK, Xia S. Variational autoencoders for deforming 3d mesh models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, 5841–5850.

[98] Gao L, Wu T, Yuan YJ, Lin MX, Lai YK, Zhang H. Tm-net: Deep generative networks for textured meshes. *ACM Transactions on Graphics (TOG)*, 2021, 40(6): 1–15.

[99] Jimenez Rezende D, Eslami S, Mohamed S, Battaglia P, Jaderberg M, Heess N. Unsupervised learning of 3d structure from images. In *Proceedings of the 29th International Conference on Neural Information Processing Systems (NeurIPS)*, 2016, 4997–5005.

[100] Henderson P, Tsiminaki V, Lampert CH. Leveraging 2d data to learn textured 3d mesh generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, 7498–7507.

[101] Nash C, Ganin Y, Eslami SA, Battaglia P. Polygen: An autoregressive generative model of 3d meshes. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020, 7220–7229.

[102] Park JJ, Florence P, Straub J, Newcombe R, Lovegrove S. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 165–174.

[103] Mescheder L, Oechsle M, Niemeyer M, Nowozin S, Geiger A. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 4460–4470.

[104] Xu Q, Wang W, Ceylan D, Mech R, Neumann U. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, 2019, 490–500.

[105] Liu S, Saito S, Chen W, Li H. Learning to infer implicit surfaces without 3d supervision. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, 2019, 8293–8304.

[106] Peng S, Niemeyer M, Mescheder L, Pollefeys M, Geiger A. Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, 523–540.

[107] Liu SL, Guo HX, Pan H, Wang PS, Tong X, Liu Y. Deep implicit moving least-squares functions for 3D reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 1788–1797.

[108] Chibane J, Alldieck T, Pons-Moll G. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, 6970–6981.

[109] Li M, Zhang H. D2im-net: Learning detail disentangled

implicit fields from single images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 10246–10255.

[110] Poursaeed O, Fisher M, Aigerman N, Kim VG. Coupling explicit and implicit surface representations for generative 3d modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, 667–683.

[111] Jiang C, Sud A, Makadia A, Huang J, Nießner M, Funkhouser T, et al.. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, 6001–6010.

[112] Jiang C, Marcus P, et al.. Hierarchical detail enhancing mesh-based shape generation with 3d generative adversarial network. *arXiv preprint arXiv:1709.07581*, 2017.

[113] Chen Z, Zhang H. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 5939–5948.

[114] Ibing M, Lim I, Kobbelt L. 3d shape generation with grid-based implicit functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 13559–13568.

[115] Mezghanni M, Boulkenafed M, Lieutier A, Ovsjanikov M. Physically-aware generative network for 3d shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 9330–9341.

[116] Tang JH, Chen W, Yang J, Wang B, Liu S, Yang B, Gao L. Octfield: Hierarchical implicit functions for 3d modeling. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, 2021, 12648–12660.

[117] Sanghi A, Chu H, Lambourne JG, Wang Y, Cheng CY, Fumero M, Malekshan KR. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, 18603–18613.

[118] Yan X, Lin L, Mitra NJ, Lischinski D, Cohen-Or D, Huang H. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, 6239–6249.

[119] Liu Z, Wang Y, Qi X, Fu CW. Towards Implicit Text-Guided 3D Shape Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, 17896–17906.

[120] Zou C, Yumer E, Yang J, Ceylan D, Hoiem D. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, 900–909.

[121] Schor N, Katzir O, Zhang H, Cohen-Or D. CompoNet: Learning to generate the unseen by part synthesis and composition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, 8759–8768.

[122] Wu R, Zhuang Y, Xu K, Zhang H, Chen B. Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, 829–838.

[123] Yin K, Chen Z, Chaudhuri S, Fisher M, Kim VG, Zhang H. Coalesce: Component assembly by learning to synthesize connections. In *Proceedings of 2020 International Conference on 3D Vision (3DV)*, 2020, 61–70.

[124] Kawana Y, Mukuta Y, Harada T. Neural star domain as primitive representation. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*, 2020, 7875–7886.

[125] Li J, Xu K, Chaudhuri S, Yumer E, Zhang H, Guibas L. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 2017, 36(4): 1–14.

[126] Wang H, Schor N, Hu R, Huang H, Cohen-Or D, Huang H. Global-to-local generative model for 3d shapes. *ACM Transactions on Graphics (TOG)*, 2018, 37(6): 1–10.

[127] Nash C, Williams CK. The shape variational autoencoder: A deep generative model of part-segmented 3D objects. In *Computer Graphics Forum*, volume 36, 2017, 1–12.

[128] Wu Z, Wang X, Lin D, Lischinski D, Cohen-Or D, Huang H. Sagnet: Structure-aware generative network for 3d-shape modeling. *ACM Transactions on Graphics (TOG)*, 2019, 38(4): 1–14.

[129] Mo K, Guerrero P, Yi L, Su H, Wonka P, Mitra N, Guibas LJ. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics (TOG)*, 2019, 38(6): 1–19.

[130] Gao L, Yang J, Wu T, Yuan YJ, Fu H, Lai YK, Zhang H. SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)*, 2019, 38(6): 1–15.

[131] Yang J, Mo K, Lai YK, Guibas LJ, Gao L. DSG-Net: Learning disentangled structure and geometry for 3D shape generation. *ACM Transactions on Graphics (TOG)*, 2022, 42(1): 1–17.

[132] Jones RK, Barton T, Xu X, Wang K, Jiang E, Guerrero P, Mitra NJ, Ritchie D. Shapeassembly: Learning to generate programs for 3d shape structure synthesis. *ACM Transactions on Graphics (TOG)*, 2020, 39(6): 1–20.

[133] Kalogerakis E, Chaudhuri S, Koller D, Koltun V. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics (TOG)*, 2012, 31(4): 1–11.

[134] Kim VG, Li W, Mitra NJ, Chaudhuri S, DiVerdi S, Funkhouser T. Learning part-based templates from large collections of 3D shapes. *ACM Transactions on Graphics (TOG)*, 2013, 32(4): 1–12.

[135] Huang H, Kalogerakis E, Marlin B. Analysis and synthesis of 3D shape families via deep-learned generative models of surfaces. In *Computer Graphics Forum*, volume 34, 2015, 25–38.

[136] Sung M, Su H, Kim VG, Chaudhuri S, Guibas L. ComplementMe: Weakly-supervised component suggestions for

3D modeling. *ACM Transactions on Graphics (TOG)*, 2017, 36(6): 1–12.

[137] Chen Z, Tagliasacchi A, Zhang H. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, 45–54.

[138] Paschalidou D, Katharopoulos A, Geiger A, Fidler S. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 3204–3215.

[139] Xiao YP, Lai YK, Zhang FL, Li C, Gao L. A survey on deep geometry learning: From a representation perspective. *Computational Visual Media*, 2020, 6(2): 113–133.

[140] Li R, Li X, Heng PA, Fu CW. Pointaugment: an auto-augmentation framework for point cloud classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, 6378–6387.

[141] Guo MH, Cai JX, Liu ZN, Mu TJ, Martin RR, Hu SM. PCT: Point cloud transformer. *Computational Visual Media*, 2021, 7(2): 187–199.

[142] Huang SS, Ma ZY, Mu TJ, Fu H, Hu SM. Supervoxel convolution for online 3d semantic segmentation. *ACM Transactions on Graphics (TOG)*, 2021, 40(3): 1–15.

[143] Huang J, Wang H, Birdal T, Sung M, Arrigoni F, Hu SM, Guibas LJ. Multibodysync: Multi-body segmentation and motion estimation via 3d scan synchronization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 7108–7118.

[144] Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 2017, 34(4): 18–42.

[145] Maturana D, Scherer S. 3d convolutional neural networks for landing zone detection from lidar. In *Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, 3471–3478.

[146] Maturana D, Scherer S. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Proceedings of 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, 922–928.

[147] Qi CR, Su H, Mo K, Guibas LJ. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, 652–660.

[148] Li Y, Bu R, Sun M, Wu W, Di X, Chen B. Pointcnn: Convolution on x-transformed points. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2018, 828–838.

[149] Hanocka R, Hertz A, Fish N, Giryes R, Fleishman S, Cohen-Or D. Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)*, 2019, 38(4): 1–12.

[150] Yuan YJ, Lai YK, Yang J, Duan Q, Fu H, Gao L. Mesh variational autoencoders with edge contraction pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020, 274–275.

[151] Hu SM, Liu ZN, Guo MH, Cai JX, Huang J, Mu TJ, Martin RR. Subdivision-based mesh convolution networks. *ACM Transactions on Graphics (TOG)*, 2022, 41(3): 1–16.

[152] Lorensen WE, Cline HE. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, 1987, 163–169.

[153] Hinton GE, Zemel R. Autoencoders, minimum description length and Helmholtz free energy. In *Proceedings of the 7th International Conference on Neural Information Processing Systems (NeurIPS)*, 1993, 3–10.

[154] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS)*, 2014, 2672–2680.

[155] Kingma DP, Welling M. Auto-Encoding Variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[156] Dinh L, Krueger D, Bengio Y. NICE: Non-linear Independent Components Estimation. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, 2015.

[157] Dinh L, Sohl-Dickstein J, Bengio S. Density estimation using Real NVP. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[158] Kingma DP, Dhariwal P. Glow: Generative flow with invertible 1x1 convolutions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2018, 10236–10245.

[159] Rezende D, Mohamed S. Variational inference with normalizing flows. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015, 1530–1538.

[160] Chen RT, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2018, 6572–6583.

[161] Grathwohl W, Chen RT, Bettencourt J, Sutskever I, Duvenaud D. FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[162] Edwards H, Storkey AJ. Towards a Neural Statistician. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[163] Riegler G, Osman Ulusoy A, Geiger A. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, 3577–3586.

[164] Wang PS, Liu Y, Guo YX, Sun CY, Tong X. O-cnn: Octree-

based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 2017, 36(4): 1–11.

[165] Radford A, Metz L, Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[166] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, 770–778.

[167] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017, 214–223.

[168] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC. Improved training of wasserstein gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS)*, 2017, 5767–5777.

[169] Isola P, Zhu JY, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, 1125–1134.

[170] Lazarow J, Jin L, Tu Z. Introspective neural networks for generative modeling. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, 2774–2783.

[171] Razavi A, Van den Oord A, Vinyals O. Generating diverse high-fidelity images with vq-vae-2. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, 2019, 14837–14847.

[172] Esser P, Rombach R, Ommer B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 12873–12883.

[173] Devlin J, Chang M, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL-HLT)*, 2019, 4171–4186.

[174] LeCun Y, Chopra S, Hadsell R, Ranzato M, Huang F. A tutorial on energy-based learning. *Predicting structured data*, 2006, 1(0).

[175] Xie J, Lu Y, Zhu SC, Wu Y. A theory of generative convnet. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016, 2635–2644.

[176] Qi CR, Yi L, Su H, Guibas LJ. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS)*, 2017, 5099–5108.

[177] Wang Y, Sun Y, Liu Z, Sarma SE, Bronstein MM, Solomon JM. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019, 38(5): 1–12.

[178] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS)*, 2017, 5998–6008.

[179] Zhao H, Jiang L, Jia J, Torr PH, Koltun V. Point transformer. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, 16259–16268.

[180] Pan X, Xia Z, Song S, Li LE, Huang G. 3d object detection with pointformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 7463–7472.

[181] Sederberg TW, Parry SR. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, 1986, 151–160.

[182] Navaneet K, Mandikal P, Agarwal M, Babu RV. Capnet: Continuous approximation projection for 3d point cloud reconstruction using 2d supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 2019, 8819–8826.

[183] Han Z, Chen C, Liu YS, Zwicker M. DRWR: a differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020, 3994–4005.

[184] Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, 8110–8119.

[185] Sohl-Dickstein J, Weiss E, Maheswaranathan N, Ganguli S. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015, 2256–2265.

[186] Maron H, Galun M, Aigerman N, Trope M, Dym N, Yumer E, Kim VG, Lipman Y. Convolutional neural networks on surfaces via seamless toric covers. *ACM Transactions on Graphics (TOG)*, 2017, 36(4): 71–1.

[187] Saquil Y, Xu QC, Yang YL, Hall P. Rank3DGAN: Semantic mesh generation using relative attributes. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 2020, 5586–5594.

[188] Aigerman N, Lipman Y. Orbifold Tutte embeddings. *ACM Transactions on Graphics (TOG)*, 2015, 34(6): 190–1.

[189] Bruna J, Zaremba W, Szlam A, Lecun Y. Spectral networks and locally connected networks on graphs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[190] Atwood J, Towsley D. Diffusion-convolutional neural networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems (NeurIPS)*, 2016, 1993–2001.

[191] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 29th International Conference on*

Neural Information Processing Systems (NeurIPS), 2016, 3837–3845.

[192] Qiao YL, Gao L, Rosin P, Lai YK, Chen X, et al.. Learning on 3D meshes with Laplacian encoding and pooling. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2020: 1317–1327.

[193] Feng Y, Feng Y, You H, Zhao X, Gao Y. Meshnet: Mesh neural network for 3d shape representation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 2019, 8279–8286.

[194] Liu HD, Kim VG, Chaudhuri S, Aigerman N, Jacobson A. Neural subdivision. *ACM Transactions on Graphics (TOG)*, 2020, 39(4): 124.

[195] Hu SM, Liang D, Yang GY, Yang GW, Zhou WY. Jittor: a novel deep learning framework with meta-operators and unified graph execution. *SCIENCE CHINA Information Sciences*, 2020, 63(12): 222103:1–222103:21.

[196] He K, Gkioxari G, Dollár P, Girshick R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, 2961–2969.

[197] Gregor K, Danihelka I, Graves A, Rezende D, Wierstra D. Draw: A recurrent neural network for image generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015, 1462–1471.

[198] Kato H, Ushiku Y, Harada T. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, 3907–3916.

[199] Liu S, Li T, Chen W, Li H. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, 7708–7717.

[200] Pavlakos G, Choutas V, Ghorbani N, Bolkart T, Osman AA, Tzionas D, Black MJ. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 10975–10985.

[201] Michalkiewicz M, Pontes JK, Jack D, Baktashmotlagh M, Eriksson A. Deep level sets: Implicit surface representations for 3d shape inference. *arXiv preprint arXiv:1901.06802*, 2019.

[202] Chibane J, Pons-Moll G, et al.. Neural unsigned distance fields for implicit function learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*, 2020, 21638–21652.

[203] Venkatesh R, Karmali T, Sharma S, Ghosh A, Babu RV, Jeni LA, Singh M. Deep implicit surface point prediction networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, 12653–12662.

[204] Aumentado-Armstrong T, Tsogkas S, Dickinson S, Jepson AD. Representing 3D Shapes with Probabilistic Directed Distance Fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, 19343–19354.

[205] Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, et al.. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021, 8748–8763.

[206] Schuster M, Paliwal KK. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 1997, 45(11): 2673–2681.

[207] Cho K, van Merrienboer B, Gulcehre C, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, 1724–1734.

[208] Gao L, Lai YK, Yang J, Zhang LX, Xia S, Kobbelt L. Sparse data driven mesh deformation. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2019, 27(3): 2085–2100.

[209] Sedaghat N, Zolfaghari M, Amiri E, Brox T. Orientation-boosted Voxel Nets for 3D Object Recognition. In *British Machine Vision Conference(BMVC)*, 2017, 97.1–97.13.

[210] Xiang Y, Kim W, Chen W, Ji J, Choy C, Su H, Mottaghi R, Guibas L, Savarese S. Objectnet3d: A large scale database for 3d object recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, 160–176.

[211] Song S, Yu F, Zeng A, Chang AX, Savva M, Funkhouser T. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, 1746–1754.

[212] Sun X, Wu J, Zhang X, Zhang Z, Zhang C, Xue T, Tenenbaum JB, Freeman WT. Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, 2974–2983.

[213] Mo K, Zhu S, Chang AX, Yi L, Tripathi S, Guibas LJ, Su H. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 909–918.

[214] Fu H, Jia R, Gao L, Gong M, Zhao B, Maybank S, Tao D. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 2021, 129(12): 3313–3337.

[215] Xiang Y, Mottaghi R, Savarese S. Beyond pascal: A benchmark for 3d object detection in the wild. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014, 75–82.

[216] Bronstein AM, Bronstein MM, Kimmel R. *Numerical geometry of non-rigid shapes*. 2008.

[217] Bogo F, Romero J, Loper M, Black MJ. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, 3794–3801.

[218] Song S, Lichtenberg SP, Xiao J. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), 2015, 567–576.

[219] Choi S, Zhou QY, Miller S, Koltun V. A large dataset of object scans. *arXiv preprint arXiv:1602.02481*, 2016.

[220] Welinder P, Branson S, Mita T, Wah C, Schroff F, Belongie S, Perona P. Caltech-UCSD birds 200. Technical report, Caltech, 2010.

[221] Shen T, Gao J, Yin K, Liu MY, Fidler S. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, 2021, 6087–6101.

[222] Yuan YJ, Sun YT, Lai YK, Ma Y, Jia R, Gao L. NeRF-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, 18353–18364.

[223] Liao Y, Donne S, Geiger A. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, 2916–2925.

[224] Zhang SH, Zhang SK, Xie WY, Luo CY, Yang YL, Fu H. Fast 3d indoor scene synthesis by learning spatial relation priors of objects. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2021, 28(9): 3082–3092.

[225] Qian Y, Hou J, Kwong S, He Y. PUGeo-Net: A geometry-centric network for 3D point cloud upsampling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, 752–769.

[226] Liang Y, Zhao S, Yu B, Zhang J, He F. MeshMAE: Masked Autoencoders for 3D Mesh Data Analysis. *arXiv preprint arXiv:2207.10228*, 2022.

Author biography

Qunce Xu is a postdoctoral researcher in the Department of Computer Science and Technology at Tsinghua University, Beijing, China. He received his Ph.D. degree from the University of Bath, UK, in 2021. His research interests include geometric learning, geometry processing and shape generation.

Taijiang Mu is an assistant researcher in the Department of Computer Science and Technology at Tsinghua University. He received his bachelor's degree and Ph.D., in Computer Science and Technology from Tsinghua University in 2011 and 2016, respectively. His research interests include computer graphics, visual media learning, 3D reconstruction and 3D understanding.

Yong-Liang Yang is a senior lecturer in the Department of Computer Science at the University of Bath, UK. He received the B.S. and Ph.D. degrees in Computer Science from Tsinghua University. His research area is broadly in visual computing, with particular interests in shape modeling, computational design, and interactive techniques.