Unsupervised Primitive Discovery for Improved 3D Generative Modeling

Salman H. Khan[*‡] Yulan Guo[†◇] Munawar Hayat[*⋆] Nick Barnes[‡]
[*]Inception Institute of Artificial Intelligence, UAE; [†]National University of Defense Technology, China;
[‡]Australian National University, AU; [◇]Sun Yat-sen University, China; [⋆]University of Canberra, AU
salman.khan@inceptioniai.org

Abstract

3D shape generation is a challenging problem due to the high-dimensional output space and complex part configurations of real-world objects. As a result, existing algorithms experience difficulties in accurate generative modeling of 3D shapes. Here, we propose a novel factorized generative model for 3D shape generation that sequentially transitions from coarse to fine scale shape generation. To this end, we introduce an unsupervised primitive discovery algorithm based on a higher-order conditional random field model. Using the primitive parts for shapes as attributes, a parameterized 3D representation is modeled in the first stage. This representation is further refined in the next stage by adding fine scale details to shape. Our results demonstrate improved representation ability of the generative model and better quality samples of newly generated 3D shapes. Further, our primitive generation approach can accurately parse common objects into a simplified representation.

1. Introduction

> *'The objects seen could be constructed out of parts with which we are familiar.'*
>
> L.G. Roberts

Computer vision in its early days saw the emergence of parts-based representations for object representation and scene understanding [23]. As early as 1963, Roberts [27] presented an approach to represent objects using a set of 3D polyhedral shapes. Subsequently, Guzman [10] introduced a collection of parts that appear in generic line drawings and demonstrated how they can be used to recognize 2D curved shapes. The generalized cylinders based representation to describe curved objects of Binford [3] was a significant breakthrough. It was developed further, including a pioneering contribution by Biederman, who introduced a set of basic primitives (termed as *'geons'* meaning geometrical ions) and linked it with the object recognition in human cognitive system [2].

Very recently, early research towards automatic discovery of shape primitives using deep networks have been reported in the literature. Tulsiani *et al*. [39] proposed a CNN model to predict the size and transformation parameters of primitives that were assembled together to represent generic 3D shapes. Their main draw-back is the inability to jointly represent different object categories using a single model. This requires a class-specific CNN training procedure, that is both time-consuming and difficult to scale to a large number of categories. Zou *et al*. [47] proposed a generative model based on RNNs to recover a 3D shape defined by primitives from an input depth image. Their model, however, requires primitive-level shape labellings for training, requires an accurate depth map as input and works only for a set of three related classes (i.e., chair, table and night stand).

In this work, we propose to incorporate a generic primitive based representation in the 3D generative modeling process to enhance the scalability of learned models. Our first major contribution is the automatic primitive discovery

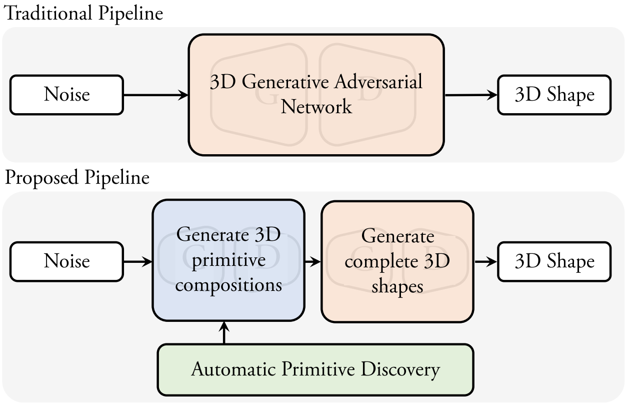

Traditional Pipeline
Noise
3D Generative Adversarial Network
3D Shape
Proposed Pipeline
Noise
Generate 3D primitive compositions
Generate complete 3D shapes
3D Shape
Automatic Primitive Discovery

Figure 1: Compared to traditional 3D generative modeling approaches (*top*) that directly generate 3D shape, our approach (*bottom*) transitions incrementally from a simple primitive based representation towards a complete 3D shape. Such a hierarchical approach provides better control and interpretability for generative networks. Furthermore, a major novelty of this work is an unsupervised primitive discovery approach that underpins the proposed generative pipeline.

in 3D shapes. Such a shape representation can provide several key benefits such as: **(a)** It factorizes the 3D generation process into a set of simpler steps, that defines a natural top-down flow in the existing bottom up generation pipelines. **(b)** It offers a highly compact representation compared to volumetric representations such as a voxel or a TSDF. **(c)** Shape primitives provide a level of abstraction in the generation process, that makes it easy to understand and manipulate the output from generative models. **(d)** A global representation of a shape encoded by a few primitives allows a better intuition about the object parts, their physical properties (e.g., stability and solidness) and their mutual relationships (e.g., support and contact) [11]. **(e)** Such a shape description provides invariance to pose changes - by explicitly estimating object size and transformation using our proposed primitive Generative Adversarial Network (GAN), the network separates viewpoint changes from the actual shape changes.

In a nutshell, we introduce the principal of modularity in the existing generative pipelines. Our main contributions are summarized below:

- A factorized generative model that improves 3D generation by introducing a simpler auxiliary task focused on learning primitive representation.
- A fully unsupervised approach based on a high-order Conditional Random Field (CRF) model to jointly optimize shape abstractions over closely related sub-sets of 3D models. Our model considers appearance, stability and physical properties of the primitives and their mutual relationships such as overlap and co-occurrence.
- The proposed model is jointly trained on all object categories and avoids expensive category specific training procedures adopted by earlier approaches.

The proposed approach can be used to incorporate intermediate levels of user input and can render more sophisticated outputs on top of that. From another perspective, it can be used to analyze the intermediate part-based representations learned by GAN that provides better interpretability and transparent generation process.

2. Related Work

3D Generative Models: Wu *et al.* [41] were the first to extend the 2D GAN framework [8] to generate 3D shapes. They demonstrated that the representations learned by the discriminator are generalizable and outperform other unsupervised classification methods. Another similar approach was proposed in [38] that used a Wasserstein loss [1] for 3D GAN. However, [38, 41] do not address primitive based shape modeling for a hierarchical shape generation pipeline. Notably, some recent efforts in 2D image generation built a hierarchy of stacked GANs to generated stage-wise outputs [13, 40, 44]. Huang *et al.* [13] used a combination of encoder, generator and discriminator blocks to perform joint top-down and bottom-up information exchange for improved image generation. However, they operate on learned feature representations and do not enhance model interpretability. Besides, a common limitation of above mentioned methods is the lack of control over the latent representations and resulting difficulties in generating data with desired attributes.

Primitive Discovery: Cuboids have been extensively used in the previous literature to represent objects, parts and scene structural elements due to their simple form [32, 18, 15, 23]. The identification of recurring parts and objects has also been studied under the problems of co-segmentation and unsupervised learning [28, 34, 29]. In 3D shapes, some efforts aim at parts discovery and modeling their mutual arrangements in large-scale shapes datasets [45, 6]. Recently, Tulsiani *et al.* [39] proposed a deep learning based approach to describe a shape with a combination of cuboid primitives. Their approach requires learning a separate model for each set of shapes belonging to the same category. Therefore, their model is not fully unsupervised and difficult to scale to a large number of object categories. In this work, we address these limitations and further propose a factorized generative model for improved shape generation.

Model Based 3D Reconstruction: The pioneering work of Roberts [27] lead to several efforts in recovering 3D layout of a scene from a single image. However, the 3D reconstruction from a single image is still an unsolved problem. Given the success of deep networks, recent approaches have proposed several incarnations of these models for 3D reconstruction. Izadinia *et al.* [14] generated 3D CAD models from a single indoor scene by detecting objects class and its pose using deep CNNs, followed by synthesizing scenes using CAD models from the ShapeNet library [4]. However, in contrast to these works, we do not have the prior knowledge about a specified set of primitives, rather we aim to automatically learn the shared parts across 3D shapes.

3. Primitive Discovery in 3D Shapes

In the first stage, we automatically discover 3D primitives from generic object shapes. Our goal is to learn common recurring primitives in 3D shapes in an unsupervised manner. We introduce a higher-order CRF model that incorporates several physical and volumetric properties of primitives to identify a consistent shape description. We propose a multi-view primitive discovery approach that discretizes the 3D space without losing much shape information and allows a computationally efficient alternative to direct 3D primitive fitting. Furthermore, since our objective is to discover shared primitives among various models, direct cuboid fitting in the original 3D space leads to more instance specific and less category generalizable primitives. Our CRF model is explained next.

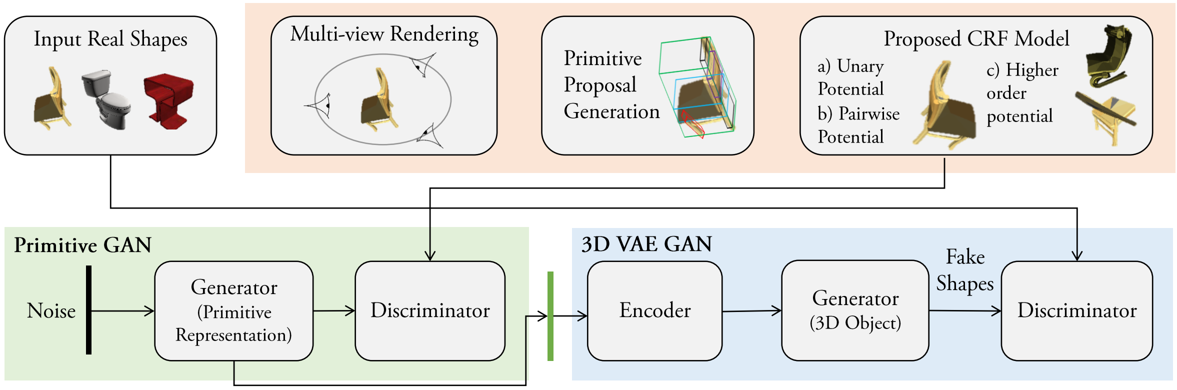
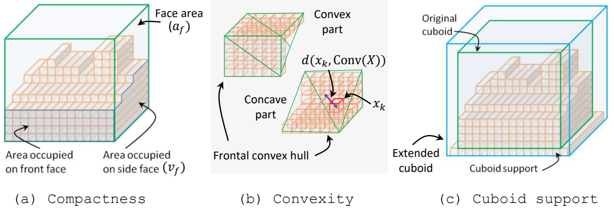


Figure 2: An overview of the proposed approach. Our model consists of a Primitive GAN that generates a parsimonious representation that is used by the 3D VAE GAN in the next stage to recover a complete 3D shape.



(a) Compactness (b) Convexity (c) Cuboid support

Figure 3: Visual illustration of costs.

3.1. Proposed CRF Model

Our goal is to automatically discover 3D primitives to represent generic 3D shapes without any supervision. For this purpose, we design a CRF model that allows efficient inference and adequately incorporates rich relationships between primitives and complete shapes. Suppose, we have a dataset $\mathcal{D} = \{\mathbf{x}_1 \dots \mathbf{x}_M\}$ consisting of M 3D shapes. For each shape $\mathbf{x}_m$, assume a candidate set of box proposals generated via bottom-up grouping (see Sec. 3.3), denoted as $\mathcal{B} = \{\mathbf{b}_1 \dots \mathbf{b}_N\}$, where N is the total number of box proposals. The segmented regions obtained by grouping are denoted by $\mathcal{R} = \{\mathbf{r}_1 \dots \mathbf{r}_R\}$. We also use a set of binary variables $\mathcal{V} = \{v_1 \dots v_N\}$ and $\mathcal{S} = \{s_1 \dots s_R\}$, where each v_i and s_r is associated with a box proposal and a segmented region, respectively. The variable v_i denotes whether a cuboid is selected as a representative primitive or not.

We develop a CRF model to encapsulate the relationships between primitives both locally as well as globally. The Gibbs energy formulation of the CRF is given by:

$$E(\mathcal{V}|\mathcal{D}) = \sum_i \psi_u(v_i) + \sum_{i<j} \psi_p(v_i, v_j) + \sum \psi_h(\mathcal{V}, \mathcal{T}),$$

where, ψ_u, ψ_p and ψ_h denote the unary, pairwise and higher-order potentials respectively and $\mathcal{T}$ represents the set of primitives from similar shapes. Next, we elaborate on each of the three potentials.

3.1.1 Unary Potential

The unary potential for each primitive candidate denotes its likelihood for a valid simplified representation of the 3D shape. This potential encodes physical and geometric properties of each box. We explain the individual cost terms within the unary potential below.

Volumetric occupancy: This cost (c_i^{oc}) estimates the empty volume within the i^{th} primitive. It is defined as $c_i^{oc} = n_i^{oc}/n_i^t$, where n_i^{oc} and n_i^t are the number of empty and total voxels respectively.

Shape uniformity: This cost (c_i^{su}) measures the uniformity of the shape along the primitive sides that were used to propose the candidate primitive. It is calculated by taking the average entropy of the surface normal direction distribution for the relevant initial segmented regions.

Primitive compactness: The cost (c_i^{pc}) estimates how tightly a 3D shape is enclosed by the primitive. It is calculated using average ratio between the empty surface area on each face and the actual face area (a_f): $c_i^{pc} = \sum_{f \in \mathcal{F}} \frac{a_f - v_f}{a_f}$, where $\mathcal{F}$ is the set of visible faces of primitive (Fig. 3).

Support cost: A valid primitive is likely to be supported by nearby shape parts. This cost calculates nearby support by considering a 5% enlarged box and taking the ratio: $c_i^{sc} = \frac{n_i^{sc}}{n_i^{ex} - n_i^{sc}}$, where n_i^{ex} and n_i^{sc} denote the number of voxels in the extended and original box primitive respectively.

Shape convexity: This cost determines the degree to which a shape-part is convex. For the regions associated with each primitive proposal, we first obtain a 3D frontal convex hull that only covers the visible 3D points from a single view. We then obtain the mean of distances between the 3D points and the frontal convex hull. It is given by: $c_i^{co} = \sum_{\forall \text{views}} \sum_k \frac{d(x_k, Conv(X))}{N}$, where, $x_k \in X$, $Conv(X)$ is the frontal convex hull for X, d denotes shortest distance between x_k and $Conv(X)$ (see Fig. 3). A large value of shape convexity cost (c_i^{co}) denotes that the shape is concave, while a small value denotes a convex shape. As convex shapes are more common in indoor scenes, a soft cost based on convexity is helpful.

Shape symmetry: For each primitive, we measure the cost (c_i^{ss}) denoting reflective symmetry of its enclosed 3D shape. For this purpose, we perform SVD decomposition to calculate three principal axis and measure the average overlap between the original points and their reflected versions. This overlap is measured as the distance between the neighboring point's position and normal direction [16]. Given the three principal orthogonal directions $X = \{a, b, c\}$ of maximum variation and the corresponding Eigenvalues denoted by π_x, the following relation is used to measure symmetry:

$$c_i^{ss} = \frac{1}{\sum\limits_x \pi_x} \sum_x \frac{\pi_x}{l_x} \Big(\sum_j \| p_j^{P_x^i} - p_{n_j}^{P_{x'}^i} \| + (1 - q_j^{P_x^i} \cdot q_{n_j}^{P_{x'}^i}) \Big),$$

where, $x{\in}X$, $j{\in}|P^i|$, P^i denotes the point cloud of i^{th} primitive, $P_{x'}$ denotes the flipped point cloud along x direction, p_j and q_j denote the j^{th} point and its normal respectively, l_x denotes the length of primitive along the x direction and n^j is the nearest neighbor of j^{th} point in $P_{x'}$.

The individual costs listed above are fused together to obtain the per primitive unary cost as follows:

$$\begin{aligned} &\psi_u(v_i) = \langle \mu_u, \mathbf{w} \circ \mathbf{c}_i \rangle, \\ &\text{where } \mathbf{c}_i = [c_i^{oc}, c_i^{su}, c_i^{pc}, c_i^{sc}, c_i^{co}, c_i^{ss}]. \end{aligned} \tag{1}$$

Here $\langle \cdot, \cdot \rangle$, $\circ$ denote inner and Hadamard products, μ_u is the cost weight vector and $\mathbf{w}$ is the normalizing vector calculated on the validation set to obtain mutually comparable costs.

3.1.2 Pairwise and High-order Potentials

Primitive Overlap: The pairwise potential considers the intersection relationships between primitive pairs. Since valid primitives do not significantly overlap each other, the goal is to penalize a configuration that violates this physical constraint. This cost c^{pw} is measured as an intersection between the two cuboids normalized by the volume of the smaller cuboid: $\psi_p(v_i, v_j) = \mu_{pw} c^{pw} v_i v_j$, where $\mu_{pw} > 0$ is the weighting parameter. In practice, we introduce an auxiliary boolean variable y_{ij} to linearize the pairwise intersection cost by replacing v_i, v_j in the above cost.

Primitive Parsimony: Motivated by the minimum description length principle, we aim to obtain a parsimonious representation of 3D shapes. In other words, we discourage using additional primatives if a small number is adequate to represent an object. A penalty on the number of active primitives is therefore introduced as a higher-order potential, $\theta_h^{par}(\mathcal{V}) = \mu^{par} \sum_{i=1}^{N} v_i,\ s.t.,\ \mu^{par} > 0$, where μ^{par} is the weight of the potential.

Coverage Potential: The minimization of costs defined above will lead to a null primitive assignment. An important requisite is to obtain a representation that maximally covers the 3D shape [39]. This constraint is formulated as maximizing the surface area enclosed by primitives:

$$\begin{aligned} &\theta_h^{cov}(\mathcal{V}, \mathcal{S}) = \mu^{cov} \sum_k c_k^{cov} s_k, \\ &s.t.,\ \mu^{cov} < 0, s_k \leq \sum_{i:\mathbf{r}_k \in \mathbf{b}_i} v_i. \end{aligned} \tag{2}$$

Here, μ_k^{cov} denotes weight and the cost c_k^{cov} is set equal to the area of the segmented region $\mathbf{r}_k$.

Co-occurrence Potential: We assume a set $\mathcal{T}$ of matched primitives for all vertices $v_i \in \mathcal{V}$. Each element $t_i = \{\hat{v}_1 \dots \hat{v}_J\} \in \mathcal{T}$ comprises of boolean variables $\hat{v}_j$ for all J primitives identified in similar shapes that are matched to primitive 'i'. The co-occurrence potential is defined as:

$$\begin{aligned} &\theta_h^{coc}(\mathcal{V}, \mathcal{T}) = \mu^{coc} \sum_{ij} c_{ij}^{coc} u_{ij}, \\ &s.t.,\ u_{ij} = v_i \hat{v}_j,\ \mu^{coc} < 0,\ v_i \leq \sum_j \hat{v}_j \end{aligned} \tag{3}$$

The variables u_{ij} and μ^{coc} denote the auxiliary boolean variable and the weight respectively. The cost c_{ij}^{coc} is defined as the Intersection over Union (IoU) measure between v_i and $\hat{v}_j$. We next describe the procedure used to find the set $\mathcal{T}$ for each 3D shape.

First, for each 3D volumetric object, a set of similar shapes is found via k-nearest neighbors in the feature space. The feature mapping is performed by obtaining a single 2D rendered image and feeding it forward through an off-the-shelf deep network [37] pre-trained on the ImageNet dataset. Afterwards, we form a complete bipartite graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ with nodes $\mathcal{N}$ and edges $\mathcal{E}$. Assume that the capacity of each edge e connecting nodes p and q is denoted by $w_{p,q} = -c_e$. The cost c_e is defined by the IoU calculated for each edge in the bipartite graph. A canonical representation is obtained for 3D shapes by aligning their principal axes and matching spatial dimensions before primitive IoU calculation. The goal is to calculate maximum weight matching $\mathcal{M}$ between the disjoint partitions $\mathcal{P}$ and $\mathcal{Q}$ defined over the m^{th} shape and its nearest neighbors. As a result, primitives within the 3D shape will have best matches that will preferably co-occur in similar 3D shapes. This problem can be formulated as an Integer Program (IP), but its solution is NP hard. To this end, we alternatively solve the following primal-dual linear relaxations of the original IP:

$$\begin{aligned} \textbf{Primal: } & \min \sum_{p,q} w_{p,q}\, y_{p,q}, \quad \text{s.t.} \sum_{p \in \mathcal{P}} y_{p,q} = 1, \\ & \sum_{q \in \mathcal{Q}} y_{p,q} = 1,\ y_{p,q} \geq 0,\ p \in \mathcal{P}, q \in \mathcal{Q}. \end{aligned} \tag{4}$$

Since the relaxed LP does not guarantee an optimal solution, we also construct a dual to the original LP where both are solved alternatively to find the optimal matching. The following lower-bound is maximized in the dual formulation:

$$\begin{aligned} \textbf{Dual:} \quad & \max \sum_{p \in \mathcal{P}} z_p + \sum_{q \in \mathcal{Q}} z_q \\ \text{s.t.} \quad & z_p + z_q \leq w_{p,q}, \quad (p,q) \in \mathcal{E} \end{aligned} \tag{5}$$

The algorithm runs in several iterations maintaining a feasible solution to the dual problem, and tries to find a feasible solution to the primal problem that satisfies complementary slackness i.e., a perfect matching $\mathcal{M}$ with only tight edges [31]. Note that if the matching is not perfect, the exposed nodes in the graph do not have corresponding co-occurrence constraints during the final optimization.

3.2. Model Inference

For a given 3D shape dataset $\mathcal{D}$, the proposed CRF model represents each shape $\mathbf{x}_m$ with a set of primitive shapes. The CRF inference is formulated as a Mixed Integer Linear Program (MILP):

$$\begin{aligned} & \mathcal{V}^* = \underset{\mathcal{V}}{\text{argmin}}\ E(\mathcal{V}|\mathcal{D}) \\ \text{s.t.} \quad & v_i = \{0,1\},\ y_{i,j} \geq 0, y_{ij} \leq v_i, y_{ij} \leq v_j, \\ & y_{ij} \geq v_i + v_j - 1,\ \ s_k \leq \sum_{i:\mathbf{r}_k \in \mathbf{b}_i} v_i, s_k \leq 1, \mu^{par} > 0, \\ & \mu^{par} > 0, \mu^{par} > 0, \mu^{cov} < 0, \mu^{coc} < 0, \\ & u_{ij} \geq 0, u_{ij} \leq v_i, u_{ij} \leq \hat{v}_j, u_{ij} \geq v_i + \hat{v}_j - 1, \\ & v_i \leq \sum_j \hat{v}_j \qquad \forall i,\ \forall i,j,\ \forall k \end{aligned} \tag{6}$$

We use branch and bound algorithm [21] to efficiently solve the MILP based inference procedure.

3.3. Primitive Proposal Generation

Here, we describe our proposed multi-view approach to generate primitive candidates. Given a polygon mesh of a 3D CAD model, we obtain rendered depth views of the model from six equi-spaced virtual viewpoints around the object. The virtual camera viewpoints were divided into two groups, one looking horizontally at the center of the upright object and the second camera viewpoint was chosen at an upward elevation of 15^0 such that the camera points towards the volume. These viewpoints were alternatively applied to get six rendered depth images that were subsequently used to obtain bottom-up polyhedron proposals. The rendered views provide sparse incomplete point clouds of the 3D shape that are mapped in the same frame of reference using a projective transformation.

As an initial step, we generate a set of 3D box proposals via bottom-up grouping. First, a normal image is calculated based on the 3D sparse point cloud for each view. Next, rough surface segmentations are obtained by clustering the 3D points that are co-located, have similar appearance and whose normals point in the same direction. Spurious segmented regions are removed by dropping regions with a small number of 3D points. We then calculate all closely lying region pairs, that can potentially form the two visible surfaces of a bounding box enclosing a part of the 3D shape. For each pair, a bounding box is tightly fit to generate a candidate primitive.

4. Generative Modeling for Shape Generation

The primitives discovered in an unsupervised manner allow us to factorize the shape generation process into two stages. The **first** GAN learns to generate novel primitive configurations that represent 3D shapes. The **second** GAN builds on this initial representation and fills in local details to generate a complete 3D shape. A Variational Auto-encoder (VAE) connects the two generative models. The overall pipeline therefore transitions from simple shape parametrization to more complex 3D shape generation. By introducing a simpler auxiliary task in the generative modeling, we achieve three key advantages: **(a)** In contrast to existing 3D generative models that are separately trained for each object category, our model is jointly trained on all shape classes, **(b)** It provides better interpretability of generator's latent space and can incorporate user input to generate desired shapes, **(c)** The learned model achieves better 3D generation results and demonstrates highly discriminative features. We explain the generative modeling pipeline below.

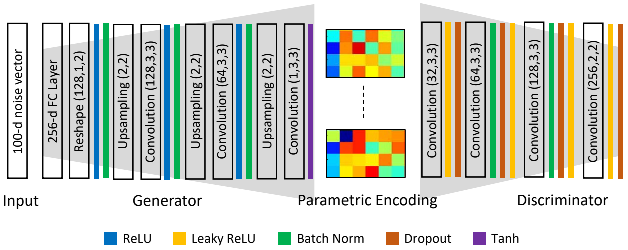


Figure 4: Primitive GAN Architecture.

4.1. Primitive GAN

Network Architecture: The primitive GAN consists of a generator and a discriminator network (Fig. 4) [17]. The training process is framed as a game between the two competing networks. The generator maps a random sample (e.g., from a Gaussian distribution) to the original data space. The discriminator operates in the data space and predicts whether an input sample is real or fake. The interesting aspect of our design is that an arbitrary number of primitives, $t \in [1, N]$ ($N = 6$ in our case), are predicted for each 3D shape. This flexibility is crucial because different object types are represented by different number of part primitives. N can be set higher at the cost of slower inference. Each primitive

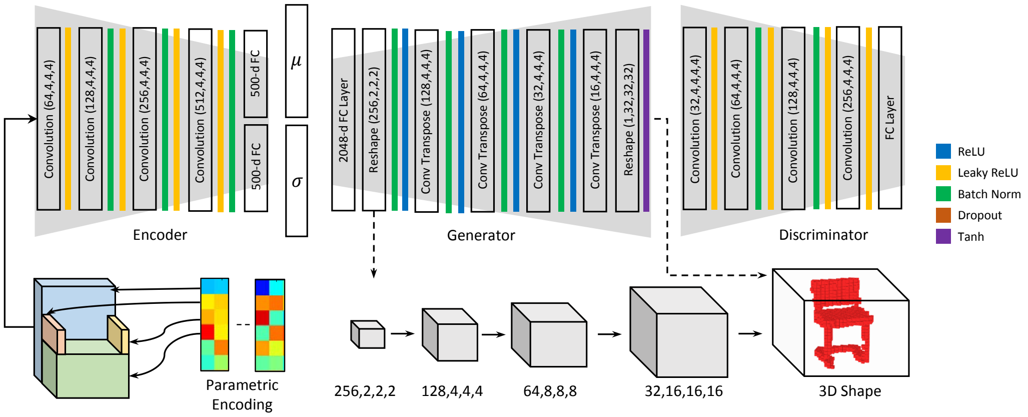


Figure 5: 3D Shape GAN Architecture. It first uses a VAE to encode the parametric representation of primitives and then learns to generate complete 3D shapes with an adversarial objective. Output tensors are shown with dotted lines.

is encoded by a shape parameter set $\theta_s \in \mathbb{R}^{15}$ including the box dimensions (i.e., height, width, depth), translation (along x, y and z axes) and the rotation matrix (with nine parameters). An additional parameter θ_l is included to denote the likelihood whether the primitive will be selected in the overall shape or not. This likelihood is used as a parameter to obtain a sample from Bernoulli distribution that denotes the existence of a primitive [39]. In this way, a 3D shape is encoded as a significantly lower dimensional parametric representation.

Loss Function: To allow a stable training of GAN, we used an improved form of Wasserstein GAN [1]. The WGAN exhibits better convergence behavior by employing Wasserstein distance as an objective. It enforces the discriminator model to remain within the space of 1-Lipschitz functions by weight clipping that can lead to sub-optimal convergence behavior. Instead of weight clipping, we used the gradient penalty introduced in [9] to restrict the norm of the gradients of the discriminator's output with respect to its input. The game between D and G is formulated as the following min-max objective function:

$$\min_G \max_D \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] - \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2], \tag{7}$$

where $\mathbb{P}_r$ is the real data distribution, $\mathbb{P}_g$ is the generator distribution modeled as $\tilde{x} = G(z)$ such that z is a random sample from a fixed distribution and $\mathbb{P}_{\hat{x}}$ is the distribution defined with uniformly sampled points between the pairs of samples belonging to $\mathbb{P}_r$ and $\mathbb{P}_g$.

4.2. 3D Shape VAE-GAN

Network Architecture: The generative model for 3D shape generation consists of a combination of a variational auto-encoder and an adversarial network (Fig. 5). The complete architecture consists of three blocks, an encoder, a generator and a discriminator. The parametric shape representation is first converted to a coarse 3D shape in the form of a voxelized grid. The encoder maps this representation to the parameters of a variational distribution by applying a series of 3D convolutional and down-sampling operations. The generator then operates on this a random sample from this parameterized distribution and generates a new 3D shape to deceive discriminator, while the discriminator is trained to correctly categorize the real and fake 3D shapes. Remarkably, in contrast to primitive GAN, the shape GAN consists of 3D operations to accurately model the data distribution of 3D shapes.

Loss Function: The loss function has the same form as for the case of primitive GAN, however, a regularization is applied on the input latent representation of generator to match it to a fixed known distribution (a unit Gaussian). This constraint is formulated as minimizing the Kullback-Leibler (KL) divergence between the Gaussian and encoded distribution as follows:

$$\mathcal{L}_{vae} = \mathrm{KL}(N(\mu,\sigma)||N(0,I)). \tag{8}$$

The reparametrization trick proposed in [19] is used to perform back-propagation through the stochastic sampling from the distribution $N(\mu,\sigma)$.

5. Experiments

Primitive Discovery: We evaluate the primitive detection accuracy on ModelNet10 dataset and report results in Table 1. Specifically, we convert the shapes and the 3D primitive representations to a 50x50x50 voxelized output. Evaluation is performed by accounting for the matched voxel predictions for both outputs. We obtain a high recall rate of 83% that confirms the correct enclosure of shape parts by primitives. In contrast, a lower precision is obtained because shape parts are often hollow, that give rise to unmatched empty voxels. In our case, recall is a more accurate measure to asses the quality of primitives. Example results for primitive generation are shown in Fig. 7.

Unsupervised Shape Classification: To illustrate the improved performance of proposed generative model, we

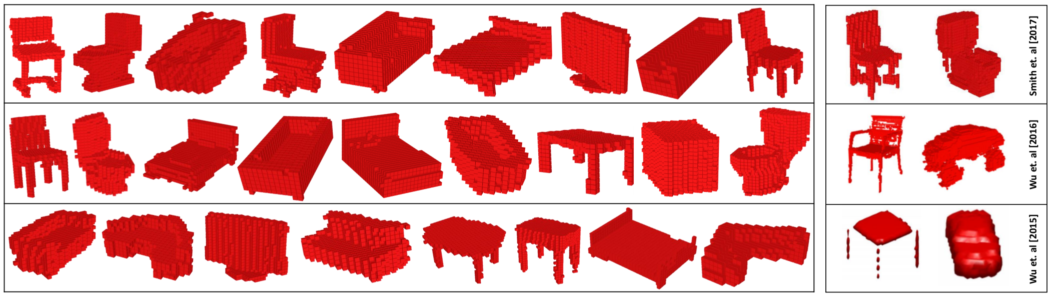


Figure 6: Qualitative results for 3D shape generation. *Left*: Generated shapes from our model. *Right:* Comparisons with [42], [41] and [38] respectively from bottom to top.

Evaluation Measure	Performance
Recall	83.0%
Precision	19.9%
Accuracy	60.8%
F-measure	0.321

Table 1: Results for the primitive generation approach on ModelNet. A high recall shows that the predicted primitives generally tightly enclose the original 3D shape.

Supervised		**Unsupervised**	
Methods	Accuracy	Methods	Accuracy
PointNet (CVPR'17) [24]	86.2%	T-L Net (ECCV'16) [7]	74.4%
OctNet (CVPR'17) [26]	83.8%	3D-GAN (NIPS'16) [41]	83.3%
Vol-CNN (Arxiv'19) [25]	86.5%	Vconv-DAE (ECCV'16)[33]	75.5%
EC-CNNs (CVPR'17)[35]	83.2%	3D-DescripNet (CVPR'18) [43]	83.8%
Kd-Net (ICCV'17) [20]	88.5%	3D-GAN (Ours)	**84.5**%
SO-Net (CVPR'18) [22]	90.8%	Primitive GAN (Ours)	**86.4**%

Table 2: Classification performance on the ModelNet40 dataset.

Type	Method	Accuracy
Supervised	3D ShapeNets (CVPR'15) [42]	93.5%
	EC-CNNs (CVPR'17) [35]	90.0%
	Kd-Net (ICCV'17) [20]	93.5%
	LightNet (3DOR'17) [46]	93.4%
	SO-Net (CVPR'18) [22]	95.5%
Unsupervised	Light Field Descriptor (CGF'03) [5]	79.9%
	Vconv-DAE (ECCV'16) [33]	80.5%
	3D-GAN (NIPS'16) [41]	91.0%
	3D-DescripNet (CVPR'18) [43]	**92.4%**
	3D-WINN (AAAI'19) [12]	91.9%
	3D-GAN (Ours)	91.2%
	Primitive GAN (Ours)	**92.2**%

Table 3: Classification performance on the ModelNet10.

Table 4: Inception scores for 3D shape generation. Best and Second-best scores are shown in color.

Method	IS
3D-ShapeNet (CVPR'15) [42]	4.13±0.19
3D-VAE (ICLR'15) [19]	11.02±0.42
3D-GAN (NIPS'16) [41]	8.66±0.45
3D-DescripNet (CVPR'18) [43]	**11.77±0.42**
3D-WINN (AAAI'19) [12]	8.81±0.18
Ours (Primitive GAN)	**11.52±0.33**

evaluate the representations learned by our discriminator (convergence plot is shown in Fig. 8). A typical way of evaluating representations learned without supervision is to use them as features for classification. Note that the Primitive GAN model is only trained in an unsupervised manner on the ModelNet10 dataset, but tested on both ModelNet10 and ModelNet40 datasets. We extract intermediate feature layers from the discriminator, concatenate them and train a single layer neural network classifier. The classification results are shown in Tables 2 and 3. Our method beats all other unsupervised techniques by a fair margin of 1.9% on ModelNet40 dataset. On ModelNet10, we achieve a competitive performance as compared to the state-of-the-art [43]. Note that some unsupervised methods have used class specific models, extra datasets (such as ShapeNet) and higher feature dimensions compared to ours. The proposed method also compares well with recent best performing fully supervised methods on both datasets. These approaches employ other tricks e.g., EC-CNNs [36] performs voting over 12 views of each test model at test time.

Inception Scores: To quantitatively evaluate the generated 3D shapes, we report Inception Score (IS) in Table 4. The IS characterizes generated objects based on two distinct criterion: the quality of 3D outputs and their diversity [30]. The quality of generated outputs is measured by the conditional probability $p(y|\mathbf{x})$, where y is the output label and $\mathbf{x}$ is the input shape. The diversity of a sample is computed by the marginal distribution $\int_z p(y|\mathbf{x} = G(z))dz$. The KL-divergence between the two gives the Inception score: IS $= \exp(\mathbb{E}[\text{KL}(p(y|\mathbf{x})||p(y))])$. Notably, using a single model for shape generation, our model achieves the second best IS score on ModelNet10 which denotes the diversity and the quality of generates shapes.

Primitive Based Reconstruction: In order to evaluate the reconstruction performance of the proposed GAN model, we test our approach on the IKEA dataset (Table 5). Previ-

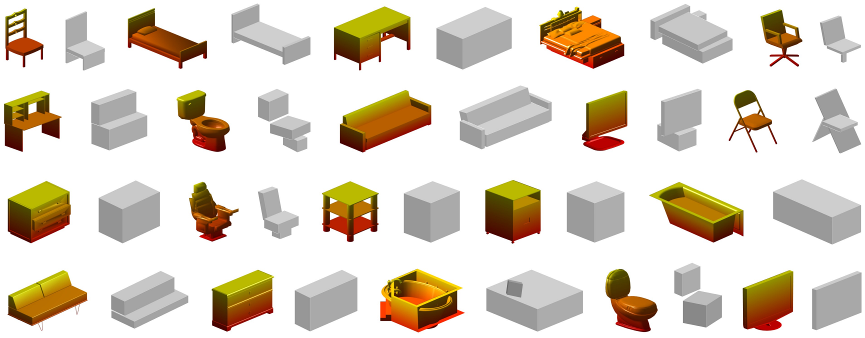


Figure 7: Automatically discovered primitive representations of 3D shapes in an unsupervised manner. Example results are shown for common indoor objects such as chair, table, desk, bathtub, sofa, monitor, toilet and nightstand. Our approach learns to represent common shapes in a parsimonious form that is consistent for examples belonging to the same category.

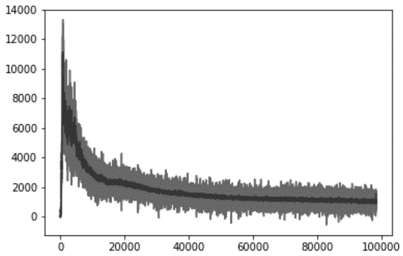


Figure 8: Discriminator loss during 3D GAN training on the ModelNet dataset.

Method	Bed	Bookcase	Chair	Desk	Sofa	Table	Overall
AlexNet-fc8† [7]	29.5	17.3	20.4	19.7	38.8	16.0	23.6
AlexNet-conv4† [7]	38.2	26.6	31.4	26.6	69.3	19.1	35.2
T-L Network† [7]	56.3	30.2	32.9	25.8	71.7	23.3	40.0
3D-VAE-GAN [41]	49.1	31.9	42.6	34.8	**79.8**	33.1	45.2
VAE-IWGAN [38]	65.7	44.2	**49.3**	50.6	68.0	52.2	55.0
Primitive GAN*	**68.4**	**52.2**	47.5	**56.9**	77.1	**60.0**	**60.4**

Table 5: Reconstruction results for voxel prediction on IKEA dataset (AP is reported). †Accuracies are reported from [41]. *Primitive GAN uses primitive representations obtained from 3D shapes and therefore has more supervision relative to compared methods that propose shapes from 2D images.

ous works e.g., [41] aim to reconstruct a 3D model from a single color image. However, in our case, the VAE-GAN model is trained on parametric inputs representing a set of basic primitives instead of image inputs. Therefore, we first run our proposed primitive discovery algorithm on the IKEA dataset to estimate primitive representations and afterwards use these to reconstruct full shapes. Note that this dataset consists of 759 images with 1039 object crops and corresponding models that belong to six objects classes namely bed, bookcase, chair, desk, sofa, and table. Since the dataset shapes are at 20x20x20, we downscale the original network output to lower resolution for evaluation.

Ablation study on cuboid detection: It is important to note that the proposed CRF formulation is an integrated framework where several potentials are optimized jointly. For example, useful primitives that are shared across similar shapes cannot be detected if we exclude any of the co-occurrence, coverage, parsimony or overlap potentials. Intuitively, one can understand that potentials like shape coverage and parsimony have opposite goals and they balance each other to get an optimal representation. However, we do run an ablation study with different types of unary potentials whose results are provided in Table 6 below. We also include a case where only unary costs are used to pick up the top four (average primitive number in dataset) primitives. We note that all potentials contribute to final performance and excluding one or some of them leads to lower recall rates.

Table 6: Ablation study on ModelNet10 for unsupervised primitive detection. The unary cost itself is insufficient to generate a good primitive representation. The best result is achieved with our full model.

Method	Recall
w/o Volumetric occupancy	72.8
w/o Shape uniformity	81.5
w/o Primitive compactness	77.2
w/o Support cost	82.9
w/o Shape symmetry	80.0
Unary only (top 4 boxes)	51.6
Full model	83.0

6. Conclusion

We factorized the generative image modeling problem to a set of simpler but interconnected tasks. Such a decomposition of problem allows GAN to generate realistic and high quality 3D voxelized representations. Our approach is motivated by the fact that common 3D objects can be represented in terms of a set of simple volumetric primitives, e.g., cuboids, spheres and cones. We first decompose a shape into a set of primitives that provide a parsimonious description with significantly less number of tunable parameters. Using this representation, we break-down the operation of GANs into simpler steps, that helps in learning better representations for data in an unsupervised fashion and makes it possible to easily incorporate user feedback if available. Such a high level supervision is helpful for complex image generation tasks such as 3D image generation and provides better interpretability and control over the outputs from GAN.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] I. Biederman. Human image understanding: Recent research and a theory. *Computer vision, graphics, and image processing*, 32(1):29–73, 1985.

[3] T. O. Binford. Visual perception by computer. In *Proceeding, IEEE Conf. on Systems and Control*, 1971.

[4] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[5] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. In *Computer graphics forum*, volume 22, pages 223–232. Wiley Online Library, 2003.

[6] N. Fish, M. Averkiou, O. Van Kaick, O. Sorkine-Hornung, D. Cohen-Or, and N. J. Mitra. Meta-representation of shape families. *ACM Transactions on Graphics (TOG)*, 33(4):34, 2014.

[7] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

[10] A. Guzmán. Decomposition of a visual scene into three-dimensional bodies. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 291–304. ACM, 1968.

[11] M. Hassanin, S. Khan, and M. Tahtali. Visual affordance and function understanding: A survey. *arXiv preprint arXiv:1807.06775*, 2018.

[12] W. Huang, B. Lai, W. Xu, and Z. Tu. 3d volumetric modeling with introspective neural networks. In *AAAI Conference on Artificial Intelligence*, 2019.

[13] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie. Stacked generative adversarial networks. In *CVPR*, 2017.

[14] H. Izadinia, Q. Shan, and S. M. Seitz. Im2cad. *arXiv preprint arXiv:1608.05137*, 2016.

[15] H. Jiang and J. Xiao. A linear approach to matching cuboids in rgbd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2171–2178, 2013.

[16] A. Karpathy, S. Miller, and L. Fei-Fei. Object discovery in 3d scenes via shape analysis. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2088–2095. IEEE, 2013.

[17] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun. A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, 8(1):1–207, 2018.

[18] S. H. Khan, X. He, M. Bennamoun, F. Sohel, and R. Togneri. Separating objects and clutter in indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4603–4611, 2015.

[19] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[20] R. Klokov and V. Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[21] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pages 497–520, 1960.

[22] J. Li, B. M. Chen, and G. H. Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[23] M. Naseer, S. Khan, and F. Porikli. Indoor scene understanding in 2.5/3d for autonomous agents: A survey. *IEEE Access*, 7:1859–1887, 2019.

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[25] S. Ramasinghe, S. Khan, and N. Barnes. Volumetric convolution: Automatic representation learning in unit ball. *arXiv preprint arXiv:1901.00616*, 2019.

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

[27] L. G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[28] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1939–1946. IEEE, 2013.

[29] J. C. Rubio, J. Serrat, A. López, and N. Paragios. Unsupervised co-segmentation through region matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 749–756. IEEE, 2012.

[30] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[31] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

[32] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3d layout and object reasoning from single images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 353–360, 2013.

[33] A. Sharma, O. Grau, and M. Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *Computer Vision–ECCV 2016 Workshops*, pages 236–250. Springer, 2016.

[34] O. Sidi, O. van Kaick, Y. Kleiman, H. Zhang, and D. Cohen-Or. *Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering*, volume 30. ACM, 2011.

[35] M. Simonovsky and N. Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[36] M. Simonovsky and N. Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3693–3702, 2017.

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[38] E. Smith and D. Meger. Improved adversarial systems for 3d object generation and reconstruction. *arXiv preprint arXiv:1707.09557*, 2017.

[39] S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros, and J. Malik. Learning shape abstractions by assembling volumetric primitives. In *Computer Vision and Pattern Regognition (CVPR)*, 2017.

[40] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision*, pages 318–335. Springer, 2016.

[41] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

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

[43] J. Xie, Z. Zheng, R. Gao, W. Wang, S.-C. Zhu, and Y. Nian Wu. Learning descriptor networks for 3d shape synthesis and analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8629–8638, 2018.

[44] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE Int. Conf. Comput. Vision (ICCV)*, pages 5907–5915, 2017.

[45] Y. Zheng, D. Cohen-Or, M. Averkiou, and N. J. Mitra. Recurring part arrangements in shape collections. In *Computer Graphics Forum*, volume 33, pages 115–124. Wiley Online Library, 2014.

[46] S. Zhi, Y. Liu, X. Li, and Y. Guo. Lightnet: A lightweight 3d convolutional neural network for real-time 3d object recognition. 2017.

[47] C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.