Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Object synthesis by learning part geometry with surface and volumetric representations

Sangpil Kim[a,*], Hyung-gun Chi[a], Karthik Ramani[a]

[a] *Purdue University, West Lafayette IN, 47906, USA*

ARTICLE INFO

Article history:
Received September 30, 2019

Keywords: Deep learning, Multi-task learning, Object synthesis, Object classification, Object reconstruction, Volumetric representation, Surface representation, Geometric

ABSTRACT

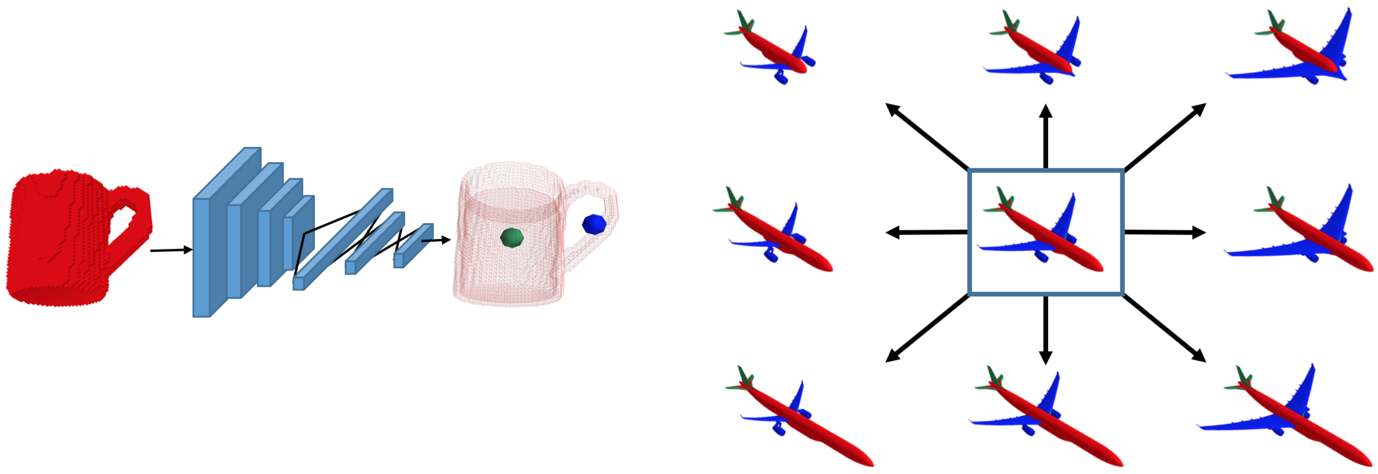
We propose a conditional generative model, named Part Geometry Network (PG-Net), which synthesizes realistic objects and can be used as a robust feature descriptor for object reconstruction and classification. Surface and volumetric representations of objects have complementary properties of three-dimensional objects. Combining these modalities is more informative than using one modality alone. Therefore, PG-Net utilizes complementary properties of surface and volumetric representations by estimating curvature, surface area, and occupancy in voxel grids of the objects with a single decoder as a multi-task learning. Objects are combinations of multiple parts, and therefore part geometry (PG) is essential to synthesize each part of the objects. PG-Net employs a part identifier to learn the part geometry. Additionally, we augmented a dataset by interpolating individual functional parts such as wings of an airplane, which helps learning part geometry and finding local/global minima of PG-Net. To demonstrate the capability of learnt object representations of PG-Net, we performed object reconstruction and classification tasks on two standard-large-scale datasets. PG-Net outperformed the state-of-the-art methods in object synthesis, classification, and reconstruction in a large margin.

1. Introduction

Technical advancement of 3D printing, virtual reality, and augmented reality has greatly increased the interest of handling three-dimensional shapes such as three-dimensional object synthesis [1, 2, 3], reconstruction [4, 5], and classification [6], which has been deeply studied in computer design communities [7, 8, 9, 10, 11]. Emergence of neural networks and creation of large-scale three-dimensional object datasets [12, 13] inspired researchers to rediscover three-dimensional object representation learning and synthesis with data-driven methods with view-based projections [14], polygon meshes [15, 16], point clouds [1, 17], and voxelized three-dimensional objects in voxel grids [18, 19]. In this work, we utilize complementary properties of surface and volumetric representations to learn features of shapes with proposed framework named Part Geometry Network (PG-Net) that synthesizes realistic objects as a conditional generative model.

Many works [20, 21, 22] have showed that jointly solving multiple tasks, named multi-task learning [23], helps improving generalizability of estimation models. From this motivation, we adopt multi-task learning to optimize PG-Net.

[*] Corresponding author: Tel.: +1-765-430-9721;
e-mail: kim2030@purdue.edu (Sangpil Kim)



(a) Part-Identifier

(b) Part Interpolation

Fig. 1: Part-Identifier and part interpolation. (a) Part-Identifier is optimized to predict the location, volume, and surface area of parts. (b) Mesh models are expanded by relocating vertices of parts that expand searching space of optimization process.

Surface representation imposes boundary and connectivity information of each part of an object [24, 25, 26], and volumetric representation determines interior geometry which is used for heat flow calculation [27, 28]. However, using these two representations for multi-task learning has not been well-examined for learning compact object representations and synthesizing objects. Surface and volumetric representations of objects contain unique features that can be complementary.

We used these modalities for multi-task learning to enhance generalizability of the model by designing the model to estimate surface and volumetric representations with the encoder-decoder structure. Knowing both surface and volumetric representations is crucial for learning not only a perceptual set of attributes but also the connectivity of each part of the objects and the details of local interior regions, hence reducing defects in synthesized objects. In this way, the model learns surface properties to learn the interior volumetric representations of objects and vice-versa [23].

Part geometry of objects is critical to learn object distribution with parametric models [29, 30, 31] since objects are combinations of specific parts. However, learning part geometry is not trivial because defining the boundary of each part is a complex problem since parts can be connected in many different ways.

Each part has their own unique shape and location for their purpose [32], which creates unique part geometry. Therefore, we propose Part-Identifier which learns part geometry for earning each part as depicted in Figure 1 a. Part-Identifier shares the part geometry information with other networks through back-propagating the meaningful gradients during training. With the understanding of part geometry, PG-Net can generate realistic objects.

However, current datasets [13, 12] have been assembled by collecting individual objects, and, therefore, each part of the objects are unique. For this reason, the datasets can not effectively share the information of parts among similar models. This makes it hard to learn connectivity between each part in the objects. To alleviate the shortage of current datasets, we expanded the dataset [33] which has part labeled triangle mesh models by reshaping specific parts of objects as shown in Figure 1 (b).

Our main contributions are summarized as follows:

1. We improved the generalizability of PG-Net by jointly estimating surface and volumetric representations as multi-task learning. In this way PG-Net learns complementary aspects of surface and volumetric properties of the object.
2. We propose a learning method of part geometry, which improves the fidelity of each part of the synthesized objects. Learning part geometry is done explicitly by optimizing

the Part-Identifier which estimates each part geometry of objects.

3. We augmented a three-dimensional object dataset to expand the learning space of PG-Net by adjusting and scaling functional parts of objects. With the expanded dataset, the optimizer can find better local/global minima of PG-Net.

We also demonstrated ablation studies and comparison experiments with other methods. We performed an object synthesis with a one-hot encoded class vector and a vector from normal distribution. For evaluating shape representations obtained from PG-Net, we performed two applications: object reconstruction and classification. From our experiments our proposed method outperformed the other state-of-the-art methods in three-dimensional object synthesis [34, 35], reconstruction [4], and classification [36, 37, 38, 1, 39, 40].

2. Related Works

Generative models have been widely studied in the field of computer vision. There are two types of well-known generative models: Variational Auto-Encoder (VAE) [41] and Generative Adversarial Network (GAN) [42]. VAE consists of an encoder and decoder. The encoder encodes input information into a low dimensional vector which is perturbed by the Gaussian noise. In GAN, the model is optimized to mimic a data distribution by playing a minimax game to find a Nash equilibrium [43] between the generator and discriminator; the generator is optimized to fool the discriminator by generating realistic data, and the discriminator is optimized to identify faked data from the generator.

Generative models with object priors: generative models which synthesize 3D objects with object priors use an encoder-decoder structure for generating objects. Achlioptas *et al.* [1] proposed the encoder-decoder structure model, which learns point cloud representation by minimizing Earth Movers Distance and Chamfer Distance with the supervision of objects' class information. However, this method cannot synthesize 3D objects from a noise distribution, and must have reference models. Umetani *et al.* [44] synthesized deformed quad meshes from reference objects by exploring the manifold of the parameterized mesh surfaces with an encoder-decoder framework. Liu *et al.* [45] proposed a method that reconstructs user inputted 3D objects by mapping them into the hidden latent space and decoding it. Xie *et al.* [2] introduced an energy-based model which approximates the 3D shape probability distribution with Markov Chain Monte Carlo methods. Groueix *et al.* [3] generated the surface of 3D shapes with a generative model from point cloud objects or images. Kalogerakis *et al.* [46] and Carlson *et al.* [47] synthesized new 3D objects by reassembling the parts which are retrieved from an existing database with non-parametric approaches. These works require reference objects to synthesize or deform objects. However, our proposed model synthesizes objects without observing objects or images as prior information.

Generative models without object priors: generative models without object priors use a decoder structure for generating synthetic objects. Wu *et al.* [38] proposed a generative adversarial loss with 3D volumetric convolution and synthesized novel 3D objects from normal distribution. Smith *et al.* [34] improved 3D-GAN with Wasserstein GAN [48], which enhances the stability of the learning process. The conditional generative adversarial network (CGAN) [49] generates targeted images given a one-hot encoded class vector and noise vectors. Chen *et al.* [19] generated colored 3D objects in voxel grids given shape descriptions by jointly learning representations of the text description and 3D colored shapes with metric learning. Conditional Variational Auto-Encoder (CVAE) [50] has been used to learn specific patterns which are structured in the underlying data distribution. Bao *et al.* [51] combined the CVAE framework and adversarial loss, which performed fine-grained image generation. The conditional generative model for 3D objects has not been well-explored, and thus, existing methods in the 3D domain have no explicit controllability to sample a specific class of 3D objects with a single pipeline. In our method, a single pipeline is used to generate targeted objects given a targeted class one-hot vector.

Object reconstruction: various methods have been proposed for object reconstruction in three-dimensional domain. Dai *et al.* [4] reconstructed a corrupted distance field and state voxel grid with an encoder-decoder model and a shape database. The final output of this method was the reconstructed distance fields of 3D objects in the 3D space. Sung *et al.* [52] used the structure of 3D objects as prior knowledge for shape completion given noisy depth scans of objects. These works were specifically dedicated to solve shape reconstruction task.

3. Preliminary

Intersected Surface Area (ISA), which calculates intersected area within a cubic voxel for all voxels in defined voxel grids, was introduced by Yarotsky [15]. ISA contains the surface area value of each voxel grid as depicted in Figure 2 a. The Mean Curvature (MC), noted as H, is the mean value of maximum and minimum curvatures , which are extrinsic measures of curvature. We also tested 2-ring and 3-ring neighborhoods of vertices for mean curvature calculation in triangle mesh models, but the result was the best when using a 1-ring neighborhood. Therefore, we used 1-ring neighborhood for mean curvature and assigned the values into voxel grids, which are illustrated in Figure 2 b.

Interior Volume (IV) is a collection of Boolean values in voxel grids in Figure 2 c. In IV, if the cubic voxel is enclosed by the surface of an object; then, the voxel value is assigned as true. These three modalities have complimentary properties, and it is listed in Table 1.

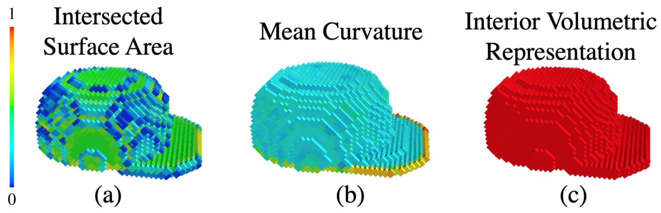


Fig. 2: Object representation modalities. (a) is the Intersected Surface Area (ISA) in a cubic voxel , (b) is the final representation in voxel grids of Mean Curvature (MC), and (c) is the Interior Volumetric (IV) representations of an object.

Properties	IV	MC	ISA
Surface area information			✓
Association of vertices		✓	
Volume information	✓		

Table 1: Complementary properties of three different modalities.

For notations in this paper, $\mathcal{D}$, D, and R, are the discriminator, decoder, and refiner respectively. Generator is the combination of the decoder and refiner. H and isa are mean curvature and intersected surface area in a voxel, respectively. pl_i, pv_i and ps_i are a $\{x, y, z\}$ Cartesian coordinate of a central location, volume and surface area, respectively, where the lower index $i \in$ {Body, Wheel, ... Legs} is the index of each part of the objects. The $\mathbf{z} \in \mathbb{R}^{200}$ is a vector of $\mu + r \odot exp(\epsilon)$, where $r \sim N(0, 1)$, N is normal distribution, and $\odot$ is an element-wise multiplication. μ and ϵ are the mean and covariance of the latent vector from the encoder. p_d is data distribution and p_z is a prior distribution of the decoder. $\mathbf{v}$ is the Boolean voxels from the true data distribution.

4. Multi-task Learning

In PG-Net, the decoder estimates three modalities as multiple tasks. PG-Net estimates MC and ISA which are surface modalities because recent works [53, 54] showed that surface properties are informative for data-driven representation learning. As shown in section 7.4, unlike a volumetric-representation-alone framework, consolidating surface knowledge penalizes inaccurate surface estimates. Therefore, we designed the decoder to estimate MC and ISA of objects along with IV, which is illustrated in Figure 3 b. We further experimented with surface normal vectors, but the result was not better than using curvature and surface area information. This is because surface normal vectors are sensitive to orientation. As one of the tasks, the decoder was optimized to learn the surface representation by minimizing the cost function $\mathcal{L}_{surf}$:

$$\sum \|\widehat{isa} - isa\|_1 + \lambda \cdot \sum \|\widehat{H} - H\|_1 \tag{1}$$

, where λ is a hyperparameter.

Another task was estimating volumetric representation of objects. PG-Net learns interior volume information of objects by minimizing sum of Sigmoid Cross-Entropy loss $\mathcal{L}_{vol}$:

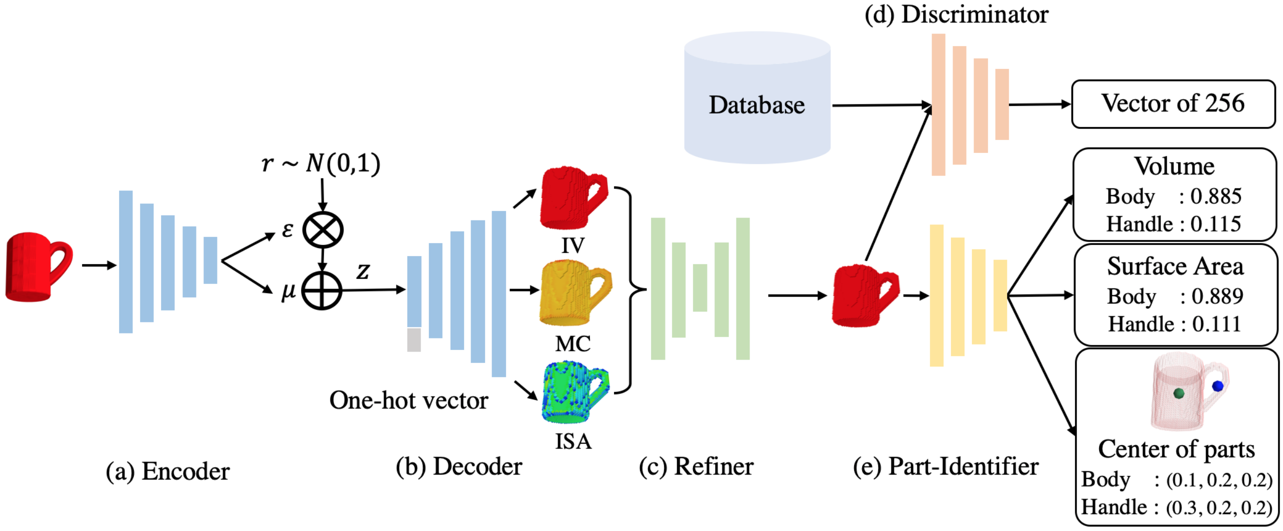


Fig. 3: Overview of PG-Net. The encoder encodes IV of the objects, and the decoder synthesizes IV, MC, and ISA. A diverse 3D object from a specified category is synthesized given a one-hot encoded class vector and a noise vector from the normal distribution. During the test stage, the encoder is removed from the pipeline. Noise vector **z** is concatenated with a one-hot vector.

$$-\mathbf{v}\log(\mathcal{D}(\mathbf{z})) - (1-\mathbf{v})\log(1-\mathcal{D}(\mathbf{z})) \tag{2}$$

In terms of the learning strategy, synthesizing IV, MC, and ISA with a single decoder solves multiple tasks. Expressed mathematically, PG-Net is optimized to jointly minimize the two cost functions:

$$\mathcal{L}_{vol} + \kappa \cdot \mathcal{L}_{surf} \tag{3}$$

,where κ is a hyperparameter. Jointly, learning both surface and volumetric representations is an informative way to discriminate objects because they provide complementary information of objects [55]. Therefore, PG-Net uses the knowledge of surface and volumetric representations for learning object distribution in a three-dimensional space.

5. Refiner and Discriminator

Adversarial training has been remarkably successful for synthesizing images and objects [56]. The goal of adversarial training is finding equilibrium between a generator and discriminator by playing a minimax game between the generator and discriminator. Motivated by adversarial training, we experimented with an adversarial loss term [42] and found that high fidelity objects are generated. To enhance the stability of adversarial training, we used the least square adversarial loss [57] $\mathcal{L}_{gan}$:

$$\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[(\mathcal{D}(R(D(\mathbf{z}))-1))^2]+\mathbb{E}_{\mathbf{v}\sim p_d}[(D(\mathbf{v})-1)^2]+\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[(\mathcal{D}(R(D(\mathbf{z}))))^2] \tag{4}$$

The discriminator discriminates fake objects from model distribution and true objects from the database. To train the discriminator, we minimized the cost function:

$$\mathbb{E}_{\mathbf{v}\sim p_d}[(D(\mathbf{v})-1)^2] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[(\mathcal{D}(R(D(\mathbf{z}))))^2] \tag{5}$$

To synthesize objects given **z**, we minimize the gap between $p(\mathbf{z})$ and three-dimensional object distribution in the latent space [51] by minimizing $\mathcal{L}_{KL}$:

$$\mu^T\mu + sum(exp(\epsilon) - \epsilon - 1) \tag{6}$$

The refiner is composed of an hourglass architecture [58] which is shown in Figure 3 c and refines coarse objects from the decoder by consolidating estimated three modalities. We found that sometimes the estimation of three modalities are not accurate or consistent. To avoid these problems, we designed the refiner to penalize poorly estimated modalities from the decoder with the loss term $\mathcal{L}_{ref}$:

$$-\mathbf{v}\log(R(D(\mathbf{z}))) - (1-\mathbf{v})\log(1-R(D(\mathbf{z}))) \tag{7}$$

For training the refiner, we jointly trained it with Part-Identifier by optimizing the cost function:

$$\alpha \cdot \mathcal{L}_{gan} + \beta \cdot \mathcal{L}_P + \gamma \cdot \mathcal{L}_{ref} + \mathcal{L}_{KL} \tag{8}$$

6. Learning Part Geometry

Part geometry is an important factor to understand three-dimensional object space [52]. From this motivation, we propose Part-Identifier for learning part geometry, which is shown in Figure 3 e. To expand the learning space of each part of an object, we further augmented the objects by interpolating the parts of the objects such as wings of an airplane.

6.1. Part-Identifier

Part-Identifier in PG-Net estimates part geometry information and the information is back-propagated through the pipeline of PG-Net. For learning the part geometry, we considered using point-wise segmentation of parts. However, it was computationally expensive since it needs to add an additional decoder module with 3D de-convolutional layers, and therefore, we regressed the center location, surface area, and the volume of each part. The position of the central coordinate, surface area, and the volume of each part are normalized by dividing the resolution of voxel grids, the surface area, and the volume of the object, respectively. Volume and surface area of parts impose meaningful information of objects such as relations of each part. Part-Identifier implicitly learns the relationship between different parts and part geometry by estimating the locations, volumes and surface areas of parts. This module guides the generator towards lower local/global minima by minimizing the $\mathcal{L}_p$:

$$\sum_i \|\widehat{pl_i} - pl_i\|_1 + \|\widehat{ps_i} - ps_i\|_1 + \|\widehat{pv_i} - pv_i\|_1 \tag{9}$$

,where $i \in$ {Body, Wheel, ... Legs}, the index of each part.

Classes	Interpolated Parts		
Airplane	Body	Wings	
Bag	Handle	Case	
Cap	Crown	Brim	
Car	Roof	Wheel	Hood
Chair	Back	Seat	Leg
Lamp	Legs	Base	Shade
Mug	Cup	Handle	
Table	Top	Leg	

Table 2: Parts for *Expanded* dataset creation.

6.2. Part expansion

The goal of part expansion which is shown in Figure 4 is to capture part geometry effectively and expand the search space for optimizing PG-Net. In order to create our dataset, named *Expanded*, we expanded Projective [33], which is a subset of the ShapeNet [12]. Projective consists of mesh models, and each part of vertices of the object is labeled. We first chose parts which are illustrated in Table 2 from each class as listed in Table 3. We removed outliers of the labeled vertices from Projective with Density-Based Spatial Clustering of Applications with Noise (DBSCAN). Additionally, we manually filtered out outliers in the dataset. Then, we adjusted the triangle mesh vertices to augment the dataset. After adjusting the triangle mesh vertices, we manually filtered out odd-looking objects. For example, we removed an object where the table top surface area is smaller than the surface area of the leg. To complete our dataset expansion, we added the models from the Scalable dataset [33], which is a subset of the ShapeNet, into the *Expanded* to increase the diversity of objects. The statistics of our dataset are detailed in Table 3.

Classes	Expanded (Ours)	Scalable [59]	Projective [33]
Airplane	36,018	2,690	500
Bag	666	76	76
Cap	477	55	55
Car	12,739	878	500
Chair	13,041	3,746	500
Lamp	13,014	1,546	500
Mug	1,629	184	184
Table	30,015	5,263	500
Total	107,599	14,438	2,815

Table 3: Statistics of *Expanded*, Scalable [59], and Projective [33].

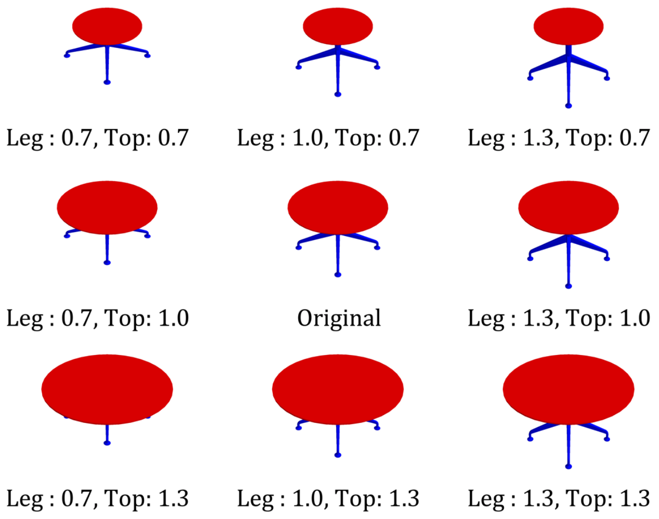


Fig. 4: Example of part expansion. Each part of tables is expanded by relocating vertices of parts to share each part information within the object. Parts are numbered based on their scaling proportion.

7. Experiments

In this section, we present experimental validations and analysis of three-dimensional object synthesis, classification, and reconstruction to validate the efficacy of PG-Net. We used two standard large-scale three-dimensional object dataset: ModelNet [13] and *Expanded*, detailed in Table 3, which is an expanded version of Projective [33]. For dataset splitting, we divided our dataset into three sets: 70% as a training set, 10% as a validation set, and 20% as a test set. We performed isosurface and Laplacian smoothing for visualizing the objects and used L_1 metric for quantitative evaluations.

7.1. Dataset preprocessing

We preprocessed mesh models to ensure the existence of mean curvature per each voxel grid. First, we voxelized the mesh models with the resolution of 40^3 voxel grids, and then we used the marching-cubes algorithm [61] to regenerate triangle meshes since voxelized objects are deformed from the original shape as depicted in Figure 6 Remeshed. We further smoothed the meshes with Laplacian smoothing [62] to reduce the variation of mean curvature as shown in Figure 6 Smoothed. From the preprocessed meshes, we calculated MC and ISA and assigned in 40^3 voxel grids. For the ISA extraction, we used SurfaceArea operation in a library [15] that computes face areas within each cubic voxel. We obtained MC by calculating the average of the minimum and maximum principal curvatures with 1-ring neighborhood distance. After obtaining all data, we normalized and scaled them to be within the interval $[-1, 1]$.

7.2. Architecture details

Designing neural network is extremely important to distinguish objects from each other and learn shape space. If the neural network is not properly designed, then the network can be biased toward the dataset which is used for the optimization of the network. In this case the network can not be used in general applications. Therefore, we carefully designed our network based on understanding the properties of the neural network.

In a neural network, salient parts are critical aspects that differentiate objects from each other. However, salient parts can have a large variation spatially in a three-dimensional space and are dependent of the kernel sizes of convolutional layers [63]. Because of this variation, choosing a right kernel size is an important factor to extract robust features from the input data. Therefore, state-of-the art image classifiers [64, 60] use filters with multiple sizes operating on the same level. From this motivation, we used Collaborate Filter (CF) which is illustrated in Figure 5 to capture the salient parts effectively from the diverse regions of the input voxels.

To capture volumetric space within voxel grid, we used 3D convolutional neural network. The squeeze and excitation block adaptively adjusts channel-wise feature responses [60]. This is effective since, in general, the neural network with multiple convolutional layers fuses channel-wise information of local receptive fields from each layer to capture meaningful features from the objects. Therefore, Up Squeeze Excitation Block (USEB) and Down Squeeze Excitation Block (DSEB) as depicted in Figure 5 can capture distinguishable features from the objects, which can be used to learn shape space and classify each objects.

Definitions of network architecture are defined in Table 4. Throughout the network, zero padding was applied when the size of the output from a previous layer is not divisible by two.

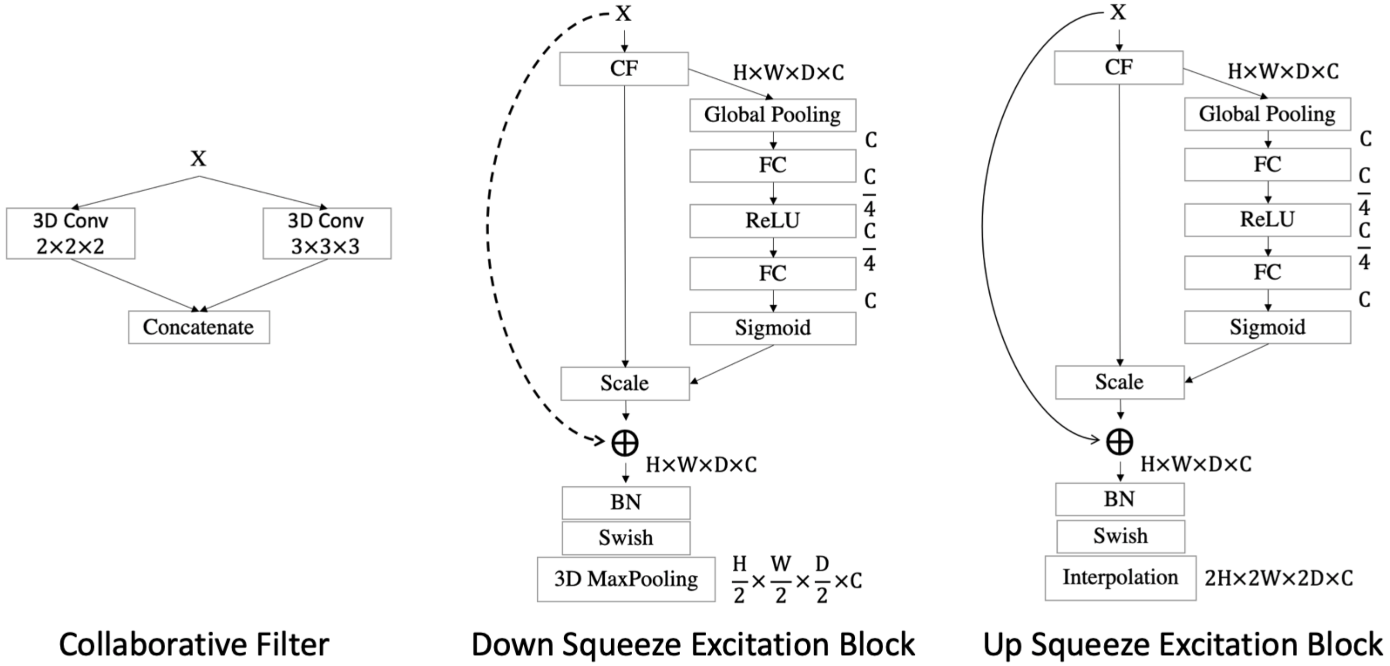


Fig. 5: Block diagrams of Collaborative Filters (CF) and Squeeze-and-Excitation block [60]. The dotted shortcuts increase dimensions if channels of X is smaller than C. BN is Batch Normalization, and C is the number of channels. Kernel size of 1×1×1 3D convolution is applied to the solid shortcut when there are miss matches in the number of channels between X and C.

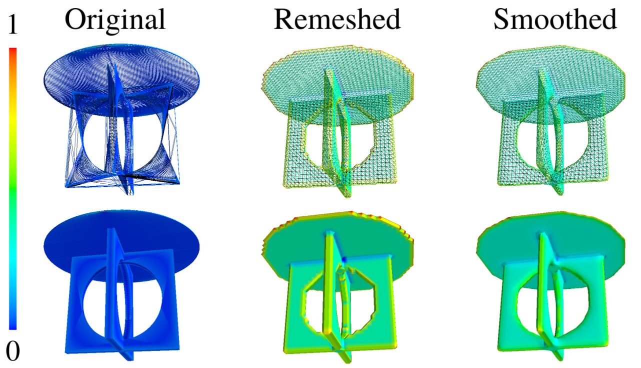


Fig. 6: Comparing an original mesh with an isosurface from a voxelized object. The first column shows mean curvatures from an original mesh. The second column shows mean curvatures from the mesh generated by marching-cubes algorithm given a voxelized object in 40^3 grids. The third column shows mean curvatures from the smoothed mesh by applying Laplacian smoothing with the second column mesh.

We used batch normalization layer and Swish [65] activation function in between the transition layers. In the discriminator, the Swish activation function was replaced with the LeakyReLu activation function on each DSEB layer.

7.3. Implementation

We used two NVIDIA TITAN Xp GPUs and an Intel i7-6850K CPU with 64GB of RAM for all of our experiments. The artificial neural network was developed with TensorFlow deep learning framework [66], which was accelerated by the CUDA instruction for the GPU computation. The networks were optimized by using the ADAM optimizer [67] with the initial parameters: learning rate = 0.0025, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. For the hyperparameters, we used $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 10^{-4}$, $\kappa = 1$, and $\gamma = 0.1$.

We trained PG-Net in two stages with a batch size of 16. In the first stage, we trained the encoder and decoder separately from other networks, which were converged approximately after 250 epochs. Then we stacked the refiner, discriminator, and Part-Identifier for the second-stage training, which requires approximately 200 epochs to converge. During the second-stage training, we did not update the encoder and decoder. In order to improve the stability of training the discriminator, we updated the refiner and Part-Identifier twice per each discriminator update to enhance the stability of the learning process [43]. After training the second-stage, we jointly updated all networks with the learning rate of 10^{-7}, which required approximately

	Type	Filter/Stride	Output Size	#Channel
A	Input		40x40x40	1
	DSEB	CF/1	20x20x20	16
	DSEB	CF/1	10x10x10	32
	DSEB	CF/1	5x5x5	64
	DSEB	CF/1	3x3x3	128
	3D-COV	1x1x1/1	2x2x2	256
	3D-COV	1x1x1/1	1x1x1	512
B	2D-COV	1x1/2	1x1	300
	BI		2x2	300
	2D-COV	3x3/2	2x2	150
	BI		5x5	150
	2D-COV	3x3/2	2x2	40
	BI		10x10	40
	Reshape		5x5x5	32
	USEB	CF/1	10x10x10	16
	USEB	CF/1	20x20x20	8
	USEB	CF/1	40x40x40	3
C	3D-COV	3x3x3/2	20x20x20	8
	3D-COV	3x3x3/2	10x10x10	16
	3D-COV	3x3x3/2	5x5x5	32
	3D-COV	3x3x3/1	5x5x5	16
	BI		10x10x10	16
	3D-COV	3x3x3/1	10x10x10	8
	BI		20x20x20	8
	3D-COV	3x3x3/1	20x20x20	1
	BI		40x40x40	1
D/E	DSEB	CF/1	20x20x20	64
	DSEB	CF/1	10x10x10	128
	DSEB	CF/1	5x5x5	256
	DSEB	CF/1	3x3x3	512
	FC		512	
	FC		256(D)/95(E)	

Table 4: Definitions of network architecture. A, B, C, D, and E are the encoder, decoder, refiner, discriminator, and Part-Identifier, respectively. In the type column, BI, FC, and COV are abbreviations of bilinear interpolation, fully-connected layer, and the convolutional layer, respectively.

100 epochs to converge. We used the initial learning rates of 0.0005 and 0.0025 for the first-stage and second-stage training, respectively. We dropped the running rate by half in every 25 epochs. For synthesizing objects, we synthesized objects from $\mathbf{z} \in \mathbb{R}^{200}$ which was sampled from a normal distribution $N(0, 1)$ and a one-hot encoded class vector.

7.4. Ablation study

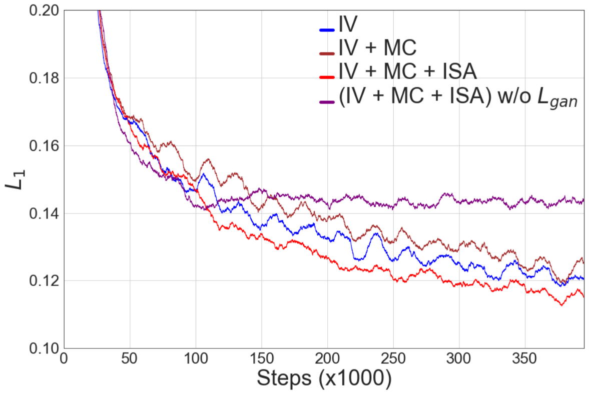


Fig. 7: L_1 loss plots of four experimental models on the test set.

	Seen	Unseen
3D-CIWGAN [34]	0.225	0.232
3D-CVAE [35]	0.173	0.199
PG-Net w/o Part-Identifier	0.164	0.194
PG-Net w/o Part Interpolation	0.183	0.204
PG-Net w/o Refiner	0.112	0.135
PG-Net (Ours)	**0.083**	**0.117**

Table 5: Quantitative results as L_1 metric of baselines and other methods for 3D object generation without 3D object references. Lower value is better.

Does multi-task and adversarial learning lead to the synthesis of better quality objects? For the ablation study, we ran four experiments: (i) IV, which only estimates volumetric representations of objects for training; (ii) IV+MC, which estimates IV and MC; (iii) IV+MC+ISA, which estimates IV, MC and ISA; (iv) (IV+MC+ISA) w/o L_{gan}, which is the same as (iii) but without L_{gan}. Figure 7 shows the performances of all experiments, quantitatively, and the qualitative results in Figure 8 indicate that the network learns a structural correlation across the local surface and volume descriptors to improve the fidelity of final outputs. This ablation study validates the rationale of multi-task and adversarial learning with IV, MC, and ISA modalities.

Why learn the geometry of parts? Furthermore, we explored the efficacy of learning part geometry as illustrated in Figure 9. Table 5 shows the quantitative evaluation of the proposed method for the following baselines: (i) *w/o Part-Identifier*, which does not have the part identifier; (ii) *w/o Part interpolation*, which is the same pipeline as PG-Net but uses dataset without data augmentation with part interpolation; (iii) *w/o Refiner*, which is trained without the refiner to evaluate the efficacy of the refiner. From the result shown in Table 5, we verified that *part identifier* improves the quality of synthesized objects since the L_1's of our *PG-Net* are lower than those of *w/o*

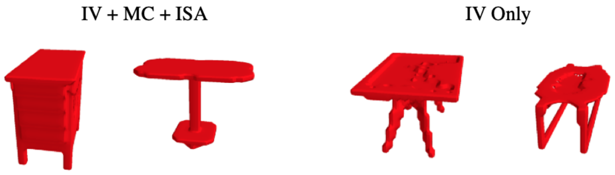


Fig. 8: Effect of the multi-tasking training with surface and volumetric representations on object synthesis. The sampled objects on the left are from the the model which was trained with IV + MC + ISA modalities, and the sampled objects on the right are from the model which was that are trained with IV only.

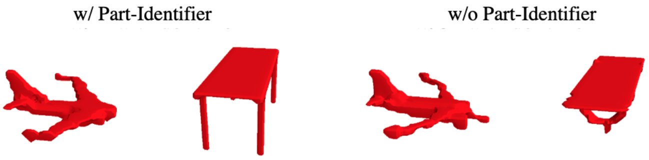


Fig. 9: Effect of the part geometric learning on object synthesis. Sampled objects from the models w/ Part-Identifier and w/o Part-Identifier given normal distribution and a one-hot encoded class vector.

Part-Identifier. Also, our PG-Net gave lower metric scores than *w/o* Part interpolation and *w/o Refiner*, which directly shows that the refiner and part interpolation results in learning robust shape representations. As a conclusion, PG-Net performs better than the other baselines, and each of our proposed methods reduces artifacts and holes of synthesized objects.

7.5. Synthesizing 3D objects

We conditionally generate 3D objects by sampling a latent vector **z**, which was mapped to the object space, and a one-hot encoded class vector. We compared our method with 3D-CVAE [35] and 3D-CIWGAN [34]. For 3D-CVAE, we followed the CVAE base model in [35] and used the same encoder/decoder definitions as PG-Net. We combined 3D-IWGAN [34] and CGAN [49] for 3D-CIWGAN. All methods used *Expanded* with a one-hot encoded class vector as a condition. Figure 10 shows the synthesized objects from our method and the others. Since we were sampling objects from noise distribution without ground truth, displaying the exact same objects for each method was impossible. Each voxel value of the synthesized objects was binarized with a threshold of 0.5. We visualized volumetric data after applying the marching-cubes algorithm and Laplacian smoothing. To assess the performance of PG-Net, we have randomly sampled objects from each class and compared the results with objects generated using alternative methods. The objects generated using PG-Net are seen to possess far fewer holes and artifacts than the objects generated using other existing methods. As shown in Figure 10, for example, both the 3D-CVAE and 3D-CIWGAN models produce objects from the ”cap” class which contain unrealistic holes. The 3D-CIWGAN results for the ”chair” class are also seen to contain a substantial amount of artifacts, particularly in columns G and H. The proposed PG-Net method does not suffer from these issues, however and is shown to produce objects from all classes which do not contain unrealistic holes or any substantial artifacts. 3D-CIWGAN is not able to synthesize the nose of the airplane, and 3D-CVAE was effective on airplanes but had many holes in the table class. Unlike 3D-CVAE and 3D-CIWGAN, PG-Net preserved part geometry and well-defined surfaces.

7.6. 3D Object classification

We evaluated object representations from PG-Net by performing 3D object classification on the ModelNet [13], which offers two kinds of datasets: ModelNet10 and ModelNet40. ModelNet10 and ModelNet40 comprises 4,899 objects and 10 classes and 12,331 objects and 40 classes, respectively. For unsupervised training, we used the same method and a dataset for fine-tuning PG-Net from Achlioptas *et al.* [1]. The dataset consists of 57,000 objects from 55 categories in ShapeNet [12]. We fine-tuned the optimized PG-Net without Part-Identifier since the parts’ labels are not available in the dataset. After fine-tuning PG-Net, we extracted features from the last two layers of encoder and concatenated them to create high-dimensional representations. Then we classified the representations with a linear SVM trained on the 3D classification benchmark [13]. Our method outperforms other existing works which are shown in Table 6. From the results, PG-Net extracts robust features which can effectively distinguish objects, and therefore, PG-Net can be used as a feature descriptor for the object analysis and other applications.

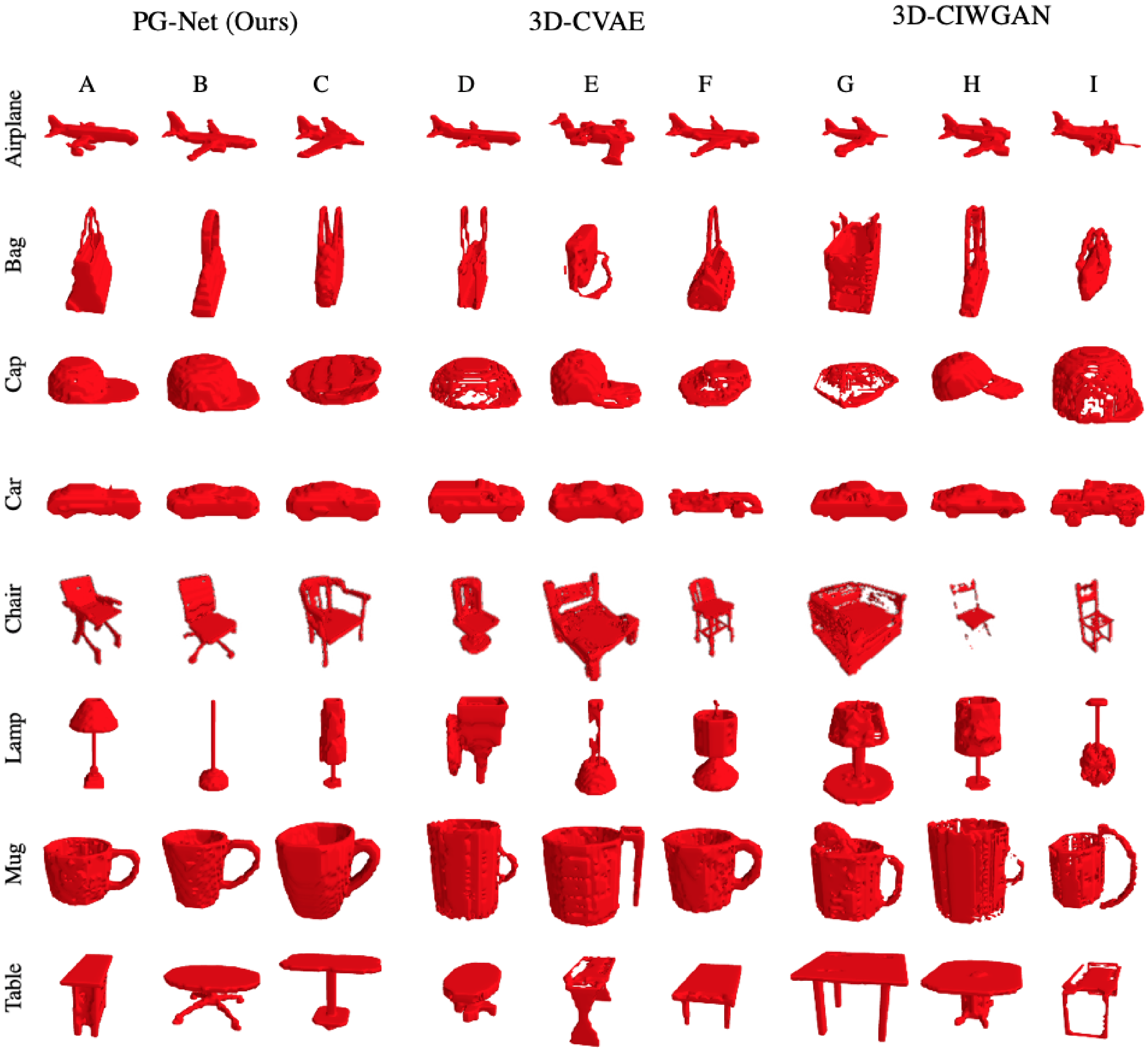


Fig. 10: Objects generated from the model given a one-hot encoded class vector and a vector from the normal distribution without a reference object. The resolution of the voxel grids was 40^3, and the objects were binarized by the threshold of 0.5.

	ModelNet10	ModelNet40
SPH [36]	79.8%	68.2%
Vconv-DAE [37]	80.5%	75.5%
ECC [39]	90.0%	83.2%
3D-GAN [38]	91.0%	83.3%
3D-DescripNet [2]	92.4%	83.8%
Primitive GAN [68]	92.2%	86.4%
FoldingNet [40]	94.4%	88.4%
GMPC [1]	95.4%	84.5%
PG-Net (Ours)	**95.6%**	**89.1%**

Table 6: The comparison on classification accuracy between PG-Net and the other unsupervised methods. Higher number is better.

Class (# of train / # of test)	3D-EPN [4]	PG-Net (Ours)
Air. (3.3K / 0.8K)	0.226	**0.202**
Car (5K / 1K)	0.197	**0.191**
Chair (5K / 1K)	0.309	**0.273**
Lamp (1.8K / 0.5K)	0.407	**0.392**
Table (5K / 1K)	0.338	**0.251**
Total (20.1K / 4.3K)	0.286	**0.249**

Table 7: Quantitative results of reconstruction task and comparison of 3D-EPN [4] and PG-Net. Lower value is better.

7.7. *Object reconstruction*

We performed three-dimensional object reconstruction to evaluate the efficacy of PG-Net and compared the results with 3D-EPN [4]. For our experiment, we used a dataset with 5 classes with around 25,000 objects from the 3D-EPN project. We interpolated the input objects from 32^3 voxel grids into 40^3 voxel grids to match the input resolution of PG-Net. For the quantitative evaluation, we converted the reconstructed objects into Boolean voxels with the threshold of 0.5. Then we counted wrongly estimated voxels and divided with the total number of occupied voxels in ground truth. We used the pre-trained weights of EPN-unet w/ class version from the 3D-EPN project page as a comparison. The quantitative results of PG-Net and 3D-EPN are compared in Table 7. The error values of PG-Net are lower than those of 3D-EPN. As shown in Figure 11, PG-Net also shows better qualitative results as compared to 3D-EPN. From quantitative and qualitative experiment results, PG-Net outperformed 3D-EPN in large margin along the all classes. Therefore, PG-Net was trained with multi-task and part geometry learning method effectively learns object distribution and nicely reconstructs three-dimensional objects which produce holes and artifacts.

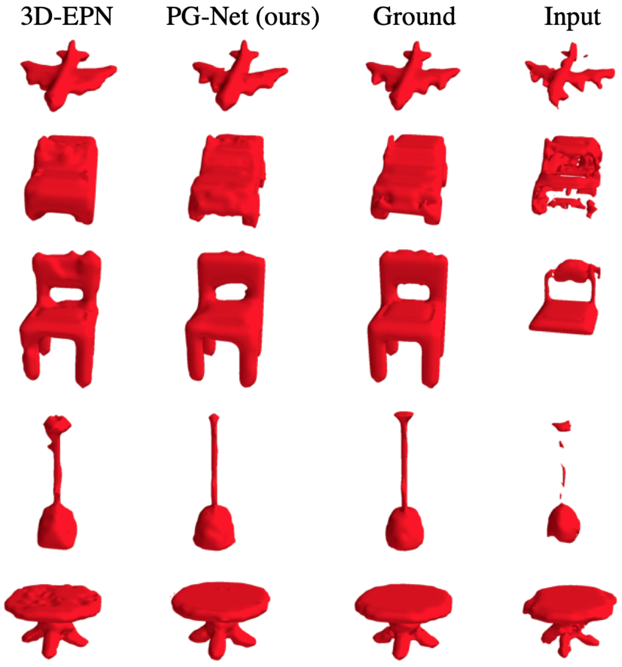


Fig. 11: Qualitative results of three-dimensional object reconstruction experiment between PG-Net and 3D-EPN.

8. Conclusions and Discussion

In this paper we propose PG-Net which was optimized with multi-task learning and part geometry learning for object synthesis. Results from our study suggest that multi-task learning increases the fidelity of generated objects and that learning part geometry enhances the realism of each part of the synthesized objects. PG-Net exceeded the other state-of-the-art methods in object synthesis, classification, and reconstruction. As limitations, 3D convolution layers with voxels are computationally expensive and thus require ample memory. However, these limitations can be solved by using a recurrent neural network with polygonal mesh vertices or point clouds representations. For a future work, fusing surface and volumetric representation as an input with point clouds representation could be further explored with a recurrent neural network.

9. Acknowledgment

This work was partially supported by the NSF under grants FW-HTF 1839971 and Convergence Accelerator 1937036 .

We also thank the Donald W. Feddersen endowment for re-initiating this research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

References

[1] Achlioptas, P, Diamanti, O, Mitliagkas, I, Guibas, L. Learning representations and generative models for 3d point clouds. In: International Conference on Machine Learning. 2018, p. 40–49.

[2] 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. 2018, p. 8629–8638.

[3] Groueix, T, Fisher, M, Kim, VG, Russell, BC, Aubry, M. Atlasnet: A papier-m\^ ach\'e approach to learning 3d surface generation. arXiv preprint arXiv:180205384 2018;.

[4] Dai, A, Ruizhongtai Qi, C, Nießner, M. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, p. 5868–5877.

[5] Choy, CB, Xu, D, Gwak, J, Chen, K, Savarese, S. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European conference on computer vision. Springer; 2016, p. 628–644.

[6] Chang, AX, Funkhouser, T, Guibas, L, Hanrahan, P, Huang, Q, Li, Z, et al. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:151203012 2015;.

[7] Hou, S, Lou, K, Ramani, K. Svm-based semantic clustering and retrieval of a 3d model database. Computer-Aided Design and Applications 2005;2(1-4):155–164.

[8] Remil, O, Xie, Q, Chen, H, Wang, J. 3d shape synthesis via content–style revealing priors. Computer-Aided Design 2019;115:87–97.

[9] Liu, Y, Pottmann, H, Wang, W. Constrained 3d shape reconstruction using a combination of surface fitting and registration. Computer-Aided Design 2006;38(6):572–583.

[10] Jayanti, S, Kalyanaraman, Y, Iyer, N, Ramani, K. Developing an engineering shape benchmark for cad models. Computer-Aided Design 2006;38(9):939–953.

[11] Shu, Z, Xin, S, Xu, H, Kavan, L, Wang, P, Liu, L. 3d model classification via principal thickness images. Computer-Aided Design 2016;78:199–208.

[12] Chang, AX, Funkhouser, T, Guibas, L, Hanrahan, P, Huang, Q, Li, Z, et al. ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR]; Stanford University — Princeton University — Toyota Technological Institute at Chicago; 2015.

[13] Wu, Z, Song, S, Khosla, A, Yu, F, Zhang, L, Tang, X, et al. 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, p. 1912–1920.

[14] Kanezaki, A, Matsushita, Y, Nishida, Y. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, p. 5010–5019.

[15] Yarotsky, D. Geometric features for voxel-based surface recognition. arXiv preprint arXiv:170104249 2017;.

[16] Masci, J, Boscaini, D, Bronstein, M, Vandergheynst, P. Geodesic convolutional neural networks on riemannian manifolds. In: Proceedings of the IEEE international conference on computer vision workshops. 2015, p. 37–45.

[17] Qi, CR, Yi, L, Su, H, Guibas, LJ. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems. 2017, p. 5099–5108.

[18] Girdhar, R, Fouhey, DF, Rodriguez, M, Gupta, A. Learning a predictable and generative vector representation for objects. In: European Conference on Computer Vision. Springer; 2016, p. 484–499.

[19] Chen, K, Choy, CB, Savva, M, Chang, AX, Funkhouser, T, Savarese, S. Text2shape: Generating shapes from natural language by learning joint embeddings. arXiv preprint arXiv:180308495 2018;.

[20] Li, S, Chan, AB. 3d human pose estimation from monocular images with deep convolutional neural network. In: Asian Conference on Computer Vision. Springer; 2014, p. 332–347.

[21] Wu, Z, Valentini-Botinhao, C, Watts, O, King, S. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2015, p. 4460–4464.

[22] Zhang, C, Zhang, Z. Improving multiview face detection with multi-task deep convolutional neural networks. In: IEEE Winter Conference on Applications of Computer Vision. IEEE; 2014, p. 1036–1041.

[23] Thrun, S, Pratt, L. Learning to learn. Springer Science & Business Media; 2012.

[24] Chen, M, Zou, Q, Wang, C, Liu, L. Edgenet: Deep metric learning for 3d shapes. Computer Aided Geometric Design 2019;72:19–33.

[25] Gao, L, Yang, J, Wu, T, Yuan, YJ, Fu, H, Lai, YK, et al. SDM-NET: Deep generative network for structured deformable mesh. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2019) 2019;38(6):To appear.

[26] Pan, H, Liu, Y, Sheffer, A, Vining, N, Li, CJ, Wang, W. Flow aligned surfacing of curve networks. ACM Transactions on Graphics (TOG) 2015;34(4):127.

[27] Kotte, A, Van Wieringen, N, Lagendijk, J. Modelling tissue heating with ferromagnetic seeds. Physics in Medicine & Biology 1998;43(1):105.

[28] Nelson, D, Charbonnel, S, Curran, A, Marttila, E, Fiala, D, Mason, P, et al. A high-resolution voxel model for predicting local tissue temperatures in humans subjected to warm and hot environments. Journal of Biomechanical Engineering 2009;131(4):041003.

[29] Mo, K, Zhu, S, Chang, AX, Yi, L, Tripathi, S, Guibas, LJ, et al. 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. 2019, p. 909–918.

[30] Yu, F, Liu, K, Zhang, Y, Zhu, C, Xu, K. Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, p. 9491–9500.

[31] Wang, X, Zhou, B, Shi, Y, Chen, X, Zhao, Q, Xu, K. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, p. 8876–8884.

[32] Yi, L, Huang, H, Liu, D, Kalogerakis, E, Su, H, Guibas, L. Deep part induction from articulated object pairs. In: SIGGRAPH Asia 2018 Technical Papers. ACM; 2018, p. 209.

[33] Kalogerakis, E, Averkiou, M, Maji, S, Chaudhuri, S. 3d shape segmentation with projective convolutional networks. In: Proc. CVPR; vol. 1. 2017, p. 8.

[34] Smith, E, Meger, D. Improved adversarial systems for 3d object generation and reconstruction. arXiv preprint arXiv:170709557 2017;.

[35] Yan, X, Yang, J, Sohn, K, Lee, H. Attribute2image: Conditional image generation from visual attributes. In: European Conference on Computer Vision. Springer; 2016, p. 776–791.

[36] Kazhdan, M, Funkhouser, T, Rusinkiewicz, S. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In: Symposium on geometry processing; vol. 6. 2003, p. 156–164.

[37] Sharma, A, Grau, O, Fritz, M. Vconv-dae: Deep volumetric shape learning without object labels. In: European Conference on Computer Vision. Springer; 2016, p. 236–250.

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

[39] Simonovsky, M, Komodakis, N. Dynamic edgeconditioned filters in convolutional neural networks on graphs. In: Proc. CVPR. 2017,.

[40] 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. 2018, p. 206–215.

[41] Kingma, DP, Welling, M. Auto-encoding variational bayes. arXiv preprint arXiv:13126114 2013;.

[42] Goodfellow, I, Pouget-Abadie, J, Mirza, M, Xu, B, Warde-Farley, D, Ozair, S, et al. Generative adversarial nets. In: Advances in neural information processing systems. 2014, p. 2672–2680.

[43] Goodfellow, I. Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:170100160 2016;.

[44] Umetani, N. Exploring generative 3d shapes using autoencoder networks. In: SIGGRAPH Asia 2017 Technical Briefs. ACM; 2017, p. 24.

[45] Liu, J, Yu, F, Funkhouser, T. Interactive 3d modeling with a generative adversarial network. arXiv preprint arXiv:170605170 2017;.

[46] Kalogerakis, E, Chaudhuri, S, Koller, D, Koltun, V. A probabilistic model for component-based shape synthesis. ACM Transactions on Graphics (TOG) 2012;31(4):55.

[47] Carlson, WE. An algorithm and data structure for 3d object synthesis using surface patch intersections. ACM SIGGRAPH Computer Graphics 1982;16(3):255–263.

[48] Arjovsky, M, Chintala, S, Bottou, L. Wasserstein gan. arXiv preprint arXiv:170107875 2017;.

[49] Mirza, M, Osindero, S. Conditional generative adversarial nets. arXiv preprint arXiv:14111784 2014;.

[50] Kingma, DP, Mohamed, S, Rezende, DJ, Welling, M. Semi-supervised learning with deep generative models. In: Advances in Neural Information Processing Systems. 2014, p. 3581–3589.

[51] Bao, J, Chen, D, Wen, F, Li, H, Hua, G. Cvae-gan: fine-grained image generation through asymmetric training. CoRR, abs/170310155 2017;5.

[52] Sung, M, Kim, VG, Angst, R, Guibas, L. Data-driven structural priors for shape completion. ACM Transactions on Graphics (TOG) 2015;34(6):175.

[53] 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):90.

[54] Kostrikov, I, Jiang, Z, Panozzo, D, Zorin, D, Bruna, J. Surface networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, p. 2540–2548.

[55] Zamir, AR, Sax, A, Shen, W, Guibas, LJ, Malik, J, Savarese, S. Taskonomy: Disentangling task transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, p. 3712–3722.

[56] Park, E, Yang, J, Yumer, E, Ceylan, D, Berg, AC. Transformation-grounded image generation network for novel 3d view synthesis. In: Proceedings of the ieee conference on computer vision and pattern recognition. 2017, p. 3500–3509.

[57] Mao, X, Li, Q, Xie, H, Lau, RY, Wang, Z, Smolley, SP. On the effectiveness of least squares generative adversarial networks. arXiv preprint arXiv:171206391 2017;.

[58] Newell, A, Yang, K, Deng, J. Stacked hourglass networks for human pose estimation. In: European conference on computer vision. Springer; 2016, p. 483–499.

[59] Yi, L, Kim, VG, Ceylan, D, Shen, I, Yan, M, Su, H, et al. A scalable active framework for region annotation in 3d shape collections. ACM Transactions on Graphics (TOG) 2016;35(6):210.

[60] Hu, J, Shen, L, Sun, G. Squeeze-and-excitation networks. arXiv preprint arXiv:170901507 2017;.

[61] Lewiner, T, Lopes, H, Vieira, AW, Tavares, G. Efficient implementation of marching cubes' cases with topological guarantees. Journal of graphics tools 2003;8(2):1–15.

[62] Vollmer, J, Mencl, R, Mueller, H. Improved laplacian smoothing of noisy surface meshes. In: Computer graphics forum; vol. 18. Wiley Online Library; 1999, p. 131–138.

[63] Paszke, A, Chaurasia, A, Kim, S, Culurciello, E. Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:160602147 2016;.

[64] Szegedy, C, Vanhoucke, V, Ioffe, S, Shlens, J, Wojna, Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, p. 2818–2826.

[65] Ramachandran, P, Zoph, B, Le, QV. Swish: a self-gated activation function. arXiv preprint arXiv:171005941 2017;.

[66] Abadi, M, Barham, P, Chen, J, Chen, Z, Davis, A, Dean, J, et al. Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). 2016, p. 265–283.

[67] Kingma, D, Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980 2014;.

[68] 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. 2019, p. 9739–9748.