EditVAE: Unsupervised Parts-Aware Controllable 3D Point Cloud Shape Generation

Shidi Li[1]**, Miaomiao Liu**[1]**, Christian Walder**[1, 2]

[1]Australian National University
[2]Data61, CSIRO
{shidi.li, miaomiao.liu}@anu.edu.au, christian.walder@data61.csiro.au

Abstract

This paper tackles the problem of parts-aware point cloud generation. Unlike existing works which require the point cloud to be segmented into parts a priori, our parts-aware editing and generation are performed in an unsupervised manner. We achieve this with a simple modification of the Variational Auto-Encoder which yields a joint model of the point cloud itself along with a schematic representation of it as a combination of shape primitives. In particular, we introduce a latent representation of the point cloud which can be decomposed into a disentangled representation for each part of the shape. These parts are in turn disentangled into both a shape primitive and a point cloud representation, along with a standardising transformation to a canonical coordinate system. The dependencies between our standardising transformations preserve the spatial dependencies between the parts in a manner that allows meaningful parts-aware point cloud generation and shape editing. In addition to the flexibility afforded by our disentangled representation, the inductive bias introduced by our joint modeling approach yields state-of-the-art experimental results on the ShapeNet dataset.

Introduction

The generation of 3D shapes has broad applications in computer graphics such as automatic model generation for artists and designers (Nash and Williams 2017), computer-aided design (Mo et al. 2020) and computer vision tasks such as recognition (Choy et al. 2015). There has been a recent boost in efforts to learn generative shape models from data (Achlioptas et al. 2018; Shu, Park, and Kwon 2019), with the main trend being to learn the distribution of 3D point clouds using deep generative models such as Variational Auto-Encoders (VAEs) (Cai et al. 2020; Yang et al. 2019), Generative Adversarial Networks (GANs) (Shu, Park, and Kwon 2019; Hui et al. 2020; Li et al. 2021), and normalising flows (Yang et al. 2019).

Recently, Mo et al. (2020) addressed structure-aware 3D shape generation, which conditions on the segmentation of point clouds into meaningful *parts* such as the legs of a chair. This yields high quality generation results, but requires time-consuming annotation of the point cloud as a *part-tree* representation. A natural alternative therefore, involves extracting

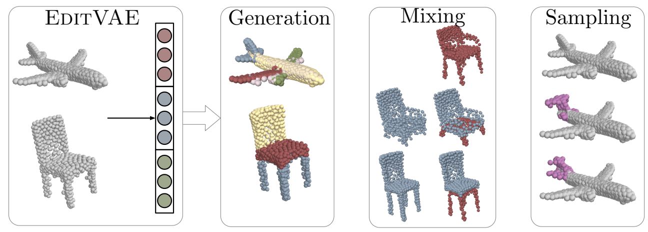

Figure 1: Our model learns a disentangled latent representation from point clouds in an unsupervised manner, allowing parts-aware generation, controllable parts mixing and parts sampling. Here we demonstrate: parts-aware *generation* as denoted by the different colours; controllable *parts mixing* to combine the legs of the upper chair with the fixed back and base of the chairs at left; and *parts sampling* of the plane stabilizers.

a semantically meaningful parts representations in an *unsupervised* manner, using ideas from recent work on *disentangled* latent representations (Chen et al. 2018; Kim and Mnih 2018)—that is, representations for which statistical dependencies between latents are discouraged. While disentanglement of the latents allows independent part sampling, reducing the dependence among parts themselves leads to samples with mis-matched style across parts.

In this paper we propose EditVAE, a framework for unsupervised parts-aware generation. EditVAE is unsupervised yet learned end-to-end, and allows parts-aware editing while respecting inter-part dependencies. We leverage a simple insight into the VAE which admits a latent space that disentangles the style and pose of the parts of the generated point clouds. Our model builds upon recent advances in primitive-based point cloud representations, to disentangle the latent space into parts, which are modeled by both latent point clouds and latent superquadric primitives, along with latent transformations to a canonical co-ordinate system. While we model point-clouds (thereby capturing detailed geometry), our model inherits from the *shape primitive* based point cloud segmentation method of Paschalidou, Ulusoy, and Geiger (2019): a semantically consistent segmentation across datasets that does not require supervision in the form of part labeling. Given the disentangled parts representation, we can perform shape editing in the space of point-clouds, e.g by exchanging the corresponding parts across point clouds or by re-sampling only some parts.

Our main contributions are summarised as follows.

1. We propose a framework for unsupervised parts-based point cloud generation.
2. We achieve reliable disentanglement of the latents by modeling points, primitives, and pose for each part.
3. We demonstrate controllable parts editing via disentangled point cloud latents for different parts.

We provide extensive experimental results on SHAPENET which quantitatively demonstrates the superior performance of our method as a generator of point clouds.

Related Work

Disentangled Latent Representation in VAE. To promote disentanglement beyond that of the vanilla VAE (Kingma and Welling 2014), Higgins et al. (2016) introduced an additional KL divergence penalty above that of the usual *evidence lower bound* (ELBO). Learning of disentangled latent representations is further investigated by Kim et. al (2018), and Chen et al. (2018). To handle minibatches while accounting for the correlation of latents, Kim et. al (2018) proposed a neural-discriminator based estimation while Chen et al. (2018) introduced a minibatch-weighted approximation. Further, Kim et al. (2019c) split latent factors into relevant and nuisance factors, treated each in a different manner within a hierarchical Bayesian model (2019b). Locatello et al. (2019a) showed that disentanglement may encourage fairness with unobserved variables, and proved the impossibility of learning disentangled representations without inductive biases (2019b) in an unsupervised manner, while showing that mild supervision may be sufficient (2020).

To learn a reliable disentangled latent representation, the present work introduces a useful inductive bias (Locatello et al. 2019b) by jointly modeling points, primitives and pose for 3D shapes. Inspired by the relevance and nuisance factor separation (Kim et al. 2019b,c), this work observes and balances the conflict between disentanglement of representation and quality of generation, by separately modeling global correlations of the relative pose of the different parts of a shape, disentangled from their style. Finally, we fill the gap of learning disentangled latent representations of 3D point cloud in an unsupervised manner, thereby contrasting with much recent disentangled representation learning works focusing on 2D or supervised cases (Kalatzis et al. 2020; Nielsen et al. 2020; Sohn, Lee, and Yan 2015).

Neural 3D Point Cloud Generation. While 2D image generation has been widely investigated using GANS (Isola et al. 2017; Zhu et al. 2017) and VAES (Kingma and Welling 2014; Higgins et al. 2016; Kim et al. 2019b; Sohn, Lee, and Yan 2015), neural 3D point cloud abstraction (Tulsiani et al. 2017; Paschalidou, Ulusoy, and Geiger 2019; Yang and Chen 2021) and generation has only been explored in recent years. Achlioptas et al. (2018) first proposed the r-GAN to generate 3D point clouds, with fully connected layers as the generator. In order to learn localized features, Valsesia et al. (2018) and Shu et al. (2019) introduced a generator based on Graph Convolutions. Specifically, Shu et al. (2019) proposed a tree-based structure with ancestors yielding a neighbor term and direct parents yielding a loop term, named the TREEGAN. This design links the geometric relationships between generated points and shared ancestors. In addition, POINTFLOW (Yang et al. 2019) learns a distribution of points based on a distribution of shapes by combining VAEs and Normalizing Flows (Rezende and Mohamed 2015), from which a point set with variable number of points may be sampled. However, all of the above works generate the point cloud as a whole or by a tree structure without disentanglement, thereby limiting their application power in parts editing. Although the work by Chen et al. (2019) focusing on reconstruction could easily be adapted to unsupervised parts-based generation task, it does not infer precise pose information which is crucial in editing.

A few recent works (Nash and Williams 2017; Mo et al. 2019, 2020; Schor et al. 2019; Dubrovina et al. 2019; Yang et al. 2021) propose (or could be adapted) to generate point clouds given ground-truth point cloud parts segmentation. However, the requirement of well-aligned parts semantic labels hinders their real world applications. MRGAN (Gal et al. 2020) firstly attempts to address the parts-aware point cloud generation by discovering parts of convex shape in an unsupervised fashion. While effective, the decomposed parts may lack semantic meaning. Following this line of work, our EDITVAE approaches parts-aware generation without semantic label requirements. In addition, the proposed model learns a disentangled latent representation, so that the style and pose of parts can be edited independently.

Preliminaries

To disentangle semantically relevant parts of a 3D point cloud, we decompose it into latent parts which are modeled both as 3D point clouds and 3D shape primitives.

A point cloud in $\mathbb{R}^{N\times 3}$ is a set of N points sampled from the surface of 3D shape in Euclidean coordinates.

Primitives are simple shapes used to assemble parts of more complex shape. We employ the superquadric parameterisation for the primitives, which is a flexible model that includes cubes, spheres and ellipsoids as special cases. In line with Paschalidou, Ulusoy, and Geiger (2019), we formally define our superquadric as the two dimensional manifold parameterised by η and ω, with surface point

$$\boldsymbol{r}(\eta,\omega) = \begin{bmatrix} \alpha_x \cos^{\epsilon_1}\eta \cos^{\epsilon_2}\omega \\ \alpha_y \cos^{\epsilon_1}\eta \sin^{\epsilon_2}\omega \\ \alpha_z \sin^{\epsilon_1}\eta \end{bmatrix} \begin{matrix} -\pi/2 \le \eta \le \pi/2 \\ -\pi \le \omega \le \pi \end{matrix}, \quad (1)$$

where $\boldsymbol{\alpha} = (\alpha_x, \alpha_y, \alpha_z)^\top$ and $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2)^\top$ are the size and shape parameters, respectively. We include additional deformation parameters based on Barr (1987) in supplementary.

Pose transformations are employed to map both the superquadric and point cloud representations of the parts from a canonical pose to the actual pose in which they appear in the *complete* point cloud. We parameterise this transformation as $\boldsymbol{x} \mapsto \mathcal{T}(\boldsymbol{x}) = \boldsymbol{R}(\boldsymbol{q})\boldsymbol{x} + \boldsymbol{t}$, which is parameterised by a translation $\boldsymbol{t} \in \mathbb{R}^3$ and a rotation defined by the quaternion $\boldsymbol{q} \in \mathbb{R}^4$. We refer to $\mathcal{T}$ as the pose for a given part.

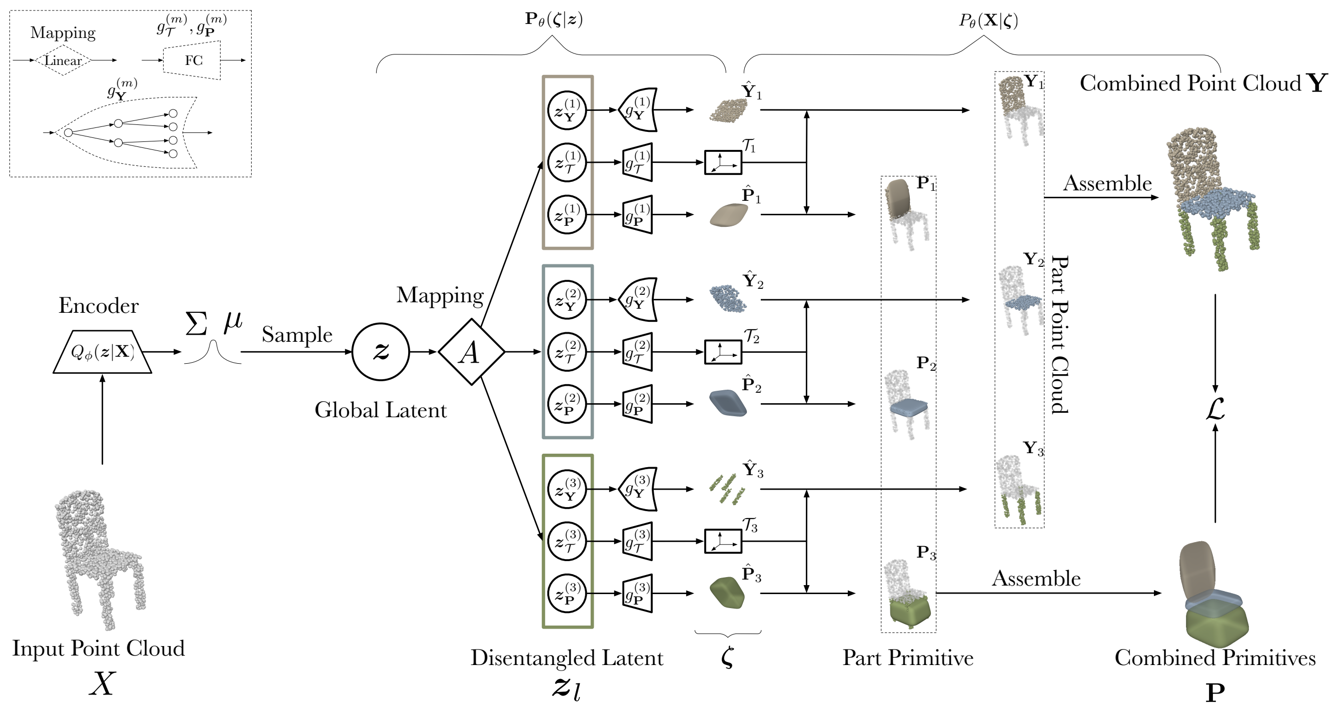

Figure 2: An overview of the EDITVAE architecture. During training, the posterior is inferred by the encoder Q_ϕ given the input point cloud X, from which a global latent $\boldsymbol{z}$ is sampled. The global latent is linearly mapped by A to the disentangled latent $\boldsymbol{z}_l$. The disentangled latent maps to parts (denoted by colors), which are further split into point $\hat{\boldsymbol{Y}}_m$, pose $\mathcal{T}_m$, and primitive $\hat{\boldsymbol{P}}_m$ representations, via the deterministic mappings $g_\star^{(i)}$. Each point $\hat{\boldsymbol{Y}}_m$ and primitive $\hat{\boldsymbol{P}}_m$ is transformed to the global coordinate system by the shared pose $\mathcal{T}_m$. The transformed part points $\boldsymbol{Y}_m$ and primitives $\boldsymbol{P}_m$ are then assembled to the complete decoded point cloud $\boldsymbol{Y}$ and primitive $\boldsymbol{P}$ models, respectively. Jointly training with a single loss $\mathcal{L}$ (far right) parsimoniously models key dependencies between point, primitive, and pose models. For generation, $\boldsymbol{z}$ is sampled from the standard Gaussian and fed forward to generate point cloud $\boldsymbol{Y}$.

Variational Auto-Encoders (VAE) (Kingma and Welling 2014) are an approximate Bayesian inference scheme that introduces an approximate posterior $Q_\phi(\boldsymbol{z}|\boldsymbol{X})$ of the latent representation conditional on the point cloud $\boldsymbol{X}$. The variational parameters ϕ are obtained by optimising a bound on the (marginal) data likelihood $P_\theta(\boldsymbol{X})$ known as the ELBO,

$$\log P_\theta(\boldsymbol{X}) \;\geq\; \mathbb{E}_{Q_\phi(\boldsymbol{z}|\boldsymbol{X})}[\log P_\theta(\boldsymbol{X}|\boldsymbol{z})] - D_{KL}\left(Q_\phi(\boldsymbol{z}|\boldsymbol{X})\right)\|P(\boldsymbol{z})) . \qquad (2)$$

The first term is known as the reconstruction error, and the second as the variational regulariser. We follow the usual approach of letting the posterior $Q_\phi(\boldsymbol{z}|\boldsymbol{X})$ be multivariate normal, so that we can employ the usual Monte Carlo approximation with the *reparameterisation trick* (Kingma and Welling 2014) to approximate the reconstruction error. By additionally letting the prior $P(\boldsymbol{z})$ be multivariate normal, we obtain a closed form expression for the regulariser.

Methodology

We motivate our design in next subsection, and then introduce our variational inference scheme, explain how we obtain disentanglement of part latents, give details of the loss functions we use, and conclude with architecture details.

Overview of the Design

We divide the challenge of parts-based point cloud generation and editing into the following essential challenges:

1. Decomposing multiple (unlabeled) point clouds into semantically meaningful parts.
2. Disentangling each part into both *style* (such as the shape of the chair leg) and the relative *pose* (the orientation in relation to the other parts of the chair).
3. Representing the components of the above decomposition in a latent space which allows style and pose to be manipulated independently of one another, while generating concrete and complete point clouds.

We address this problem in an end-to-end manner with a unified probabilistic model. To accomplish this we depart slightly from the well known VAE structure, which directly reconstructs the input by the *decoder*.

For any given input point cloud $\boldsymbol{X}$ we generate a separate point cloud $\hat{\boldsymbol{Y}}_m$ for each part m of the input point cloud (such as the base of a chair), along with a super-quadric prototype $\hat{\boldsymbol{P}}_m$ of that part. This addresses point 1 above. To address point 2, we model $\hat{\boldsymbol{P}}_m$ and $\hat{\boldsymbol{Y}}_m$ in a standardised reference pose via the affine transformation $\mathcal{T}_m$, and denote by

$$\boldsymbol{P}_m = \mathcal{T}_m(\hat{\boldsymbol{P}}_m) \quad \text{and} \quad \boldsymbol{Y}_m = \mathcal{T}_m(\hat{\boldsymbol{Y}}_m) \tag{3}$$

the point cloud and primitive part representations in the original pose. This allows a part's style to be edited while maintaining global coherance. Finally, while we model a single global latent $\boldsymbol{z}$, our decoder generates each part via separate network branches (see Figure 2), thereby facilitating various editing operations and satisfying point 3 above.

Variational Inference

Our approximate inference scheme is based on that of the VAE (Kingma and Welling 2014; Rezende, Mohamed, and Wierstra 2014), but similarly to Kim et al. (2019a) relaxes the assumption that the encoder and decoder map from and to the same (data) space. The following analysis is straightforward, yet noteworthy in that it side-steps the inconvenience of applying variational regularization to $\boldsymbol{\zeta}$.

Denote by $\zeta_m = \{\hat{\boldsymbol{Y}}_m, \hat{\boldsymbol{P}}_m, \mathcal{T}_m\}$ the m-th latent part representation, by $\boldsymbol{\zeta} = \bigcup_{m=1}^{M} \zeta_m$ the union of all such parts, and by $\boldsymbol{z}$ a global latent which abstractly represents a shape. We let $Q_\phi(\boldsymbol{z}, \boldsymbol{\zeta}|\boldsymbol{X})$ represent the approximate posterior with parameters ϕ, and for simplicity we neglect to notate the dependence of Q_ϕ on θ. Our training objective is the usual marginal likelihood of the data $\boldsymbol{X}$ given the parameters θ,

$$P_\theta(\boldsymbol{X}) = \int P_\theta(\boldsymbol{X}, \boldsymbol{z}, \boldsymbol{\zeta}) \, \mathrm{d}\boldsymbol{z} \, \mathrm{d}\boldsymbol{\zeta}. \tag{4}$$

Taking logs and applying Jensen's inequality we have

$$\begin{aligned} \log P_\theta(\boldsymbol{X}) &= \log \int P_\theta(\boldsymbol{X}, \boldsymbol{z}, \boldsymbol{\zeta}) \, \mathrm{d}\boldsymbol{z} \, \mathrm{d}\boldsymbol{\zeta} & (5) \\ &= \log \int \frac{Q_\phi(\boldsymbol{z}, \boldsymbol{\zeta}|\boldsymbol{X})}{Q_\phi(\boldsymbol{z}, \boldsymbol{\zeta}|\boldsymbol{X})} P_\theta(\boldsymbol{X}, \boldsymbol{z}, \boldsymbol{\zeta}) \, \mathrm{d}\boldsymbol{z} \, \mathrm{d}\boldsymbol{\zeta} & (6) \\ &\geq \int Q_\phi(\boldsymbol{z}, \boldsymbol{\zeta}|\boldsymbol{X}) \log \frac{P_\theta(\boldsymbol{X}, \boldsymbol{z}, \boldsymbol{\zeta})}{Q_\phi(\boldsymbol{z}, \boldsymbol{\zeta}|\boldsymbol{X})} \, \mathrm{d}\boldsymbol{z} \, \mathrm{d}\boldsymbol{\zeta}. & (7) \end{aligned}$$

We assume a chain-structured factorisation in our posterior,

$$P_\theta(\boldsymbol{z}, \boldsymbol{\zeta}|\boldsymbol{X}) = P_\theta(\boldsymbol{\zeta}|\boldsymbol{z}) P_\theta(\boldsymbol{z}|\boldsymbol{X}). \tag{8}$$

Under this factorisation we obtain a tractable variational inference scheme by assuming that conditional on $\boldsymbol{z}$, the approximate posterior matches the true one, *i.e.*

$$Q_\phi(\boldsymbol{z}, \boldsymbol{\zeta}|\boldsymbol{X}) \equiv P_\theta(\boldsymbol{\zeta}|\boldsymbol{z}) \, Q_\phi(\boldsymbol{z}|\boldsymbol{X}). \tag{9}$$

Putting (9) into (7) and cancelling $P_\theta(\boldsymbol{\zeta}|\boldsymbol{z})$ in the $\log$ in (7),

$$\begin{aligned} \log P_\theta(\boldsymbol{X}) \;\; \geq \;\; & \mathbb{E}_{Q_\phi(\boldsymbol{z}|\boldsymbol{X})} \left[\log P_\theta(\boldsymbol{X}|\boldsymbol{\zeta})\right] \\ & - D_{KL} \left(Q_\phi(\boldsymbol{z}|\boldsymbol{X}) \| P_\theta(\boldsymbol{z})\right), \end{aligned} \tag{10}$$

where $\boldsymbol{\zeta} = \mathrm{NN}_\theta(\boldsymbol{z})$. *In a nutshell,* this shows that we need only learn an approximate posterior $Q_\phi(\boldsymbol{z}|\boldsymbol{X})$ via a similar ELBO as (2), to obtain an approximate posterior on $\boldsymbol{\zeta}$. We achieve this via a simple deterministic mapping which, like Nielsen et al. (2020), we may notate as the limit $P_\theta(\boldsymbol{\zeta}|\boldsymbol{z}) = Q_\phi(\boldsymbol{\zeta}|\boldsymbol{z}) \to \delta(\boldsymbol{\zeta} - \mathrm{NN}_\theta(\boldsymbol{z}))$, where δ is the Dirac distribution and NN_θ denotes a neural network. Crucially, while the posterior in $\boldsymbol{\zeta}$ is non-Gaussian, it doesn't appear in the variational regulariser which is therefore tractable.

Disentangling the Latent Representation

EditVAE disentangles the global latent $\boldsymbol{z}$ into a local (to part ζ_m) latent $\boldsymbol{z}_l^{(i)}$, and further to latents for specific component of that part (namely $\boldsymbol{Y}_m, \boldsymbol{P}_m$ or $\mathcal{T}_m$). We achieve this key feature by *linearly* transforming and partitioning the global latent, *i.e.* we define

$$(\boldsymbol{z}_l^{(1)}, \boldsymbol{z}_l^{(2)}, \cdots, \boldsymbol{z}_l^{(M)})^\top = \boldsymbol{z}_l = A\boldsymbol{z}, \tag{11}$$

where A is a matrix of weights (representing a linear neural network layer). We further partition the part latents as

$$\boldsymbol{z}_l^{(m)} = (\boldsymbol{z}_{\boldsymbol{Y}}^{(m)}, \boldsymbol{z}_{\mathcal{T}}^{(m)}, \boldsymbol{z}_{\boldsymbol{P}}^{(m)})^\top, \tag{12}$$

and let the corresponding parts themselves be defined as

$$\hat{\boldsymbol{Y}}_m = g_{\boldsymbol{Y}}^{(m)}(\boldsymbol{z}_{\boldsymbol{Y}}^{(m)}), \tag{13}$$

and similarly for $\hat{\boldsymbol{P}}_m$ and $\mathcal{T}_m$. Here, $g_{\boldsymbol{Y}}^{(m)}$ non-linearly transforms from the latent space to the part representation.

This achieves several goals. First, we inherit from the VAE a meaningful latent structure on $\boldsymbol{z}$. Second, by *linearly* mapping from $\boldsymbol{z}$ to the local part latents $\boldsymbol{z}_{\boldsymbol{Y}}^{(i)}, \boldsymbol{z}_{\mathcal{T}}^{(i)}$ and $\boldsymbol{z}_{\boldsymbol{P}}^{(i)}$, we ensure that linear operations (*e.g.* convex combination) on the global latent precisely match linear operations on the local latent space, which therefore captures a meaningfully *local* latent structure. Finally, partitioning $\boldsymbol{z}_l$ yields a representation that disentangles parts by construction, while dependencies between parts are captured by A. Experiments show we obtain meaningful disentangled parts latents.

Loss Functions

Completing the model of the previous sub-section requires to specify the log likelihood $\log P_\theta(\boldsymbol{X}|\boldsymbol{\zeta})$, which we decompose in the usual way as the negative of a sum of loss functions involving either or both of the point $\boldsymbol{Y}_m$ and superquadric $\boldsymbol{P}_m$, representations—combined with the standardisation transformation $\mathcal{T}$ which connects these representations to the global point cloud, $\boldsymbol{X}$. Note that from a Bayesian modelling perspective, there is no need to separate the loss into terms which decouple $\boldsymbol{P}$ and $\boldsymbol{Y}$; indeed, the flexibility to couple these representations within the loss is a source of useful inductive bias in our model.

While our loss does not correspond to a normalised conditional $P_\theta(\boldsymbol{X}|\boldsymbol{\zeta})$, working with un-normalised losses is both common (Sun et al. 2019; Paschalidou, Ulusoy, and Geiger 2019), and highly convenient since we may engineer a practically effective loss function by combining various carefully designed losses from previous works.

Point Cloud Parts Loss. We include a loss term for each part point cloud $\hat{\boldsymbol{Y}}_m$ based on the Chamfer distance

$$\begin{aligned} &\mathcal{L}_c(\boldsymbol{X}, \boldsymbol{Y}) = \\ &\frac{1}{2|\boldsymbol{X}|} \sum_{x \in \boldsymbol{X}} \min_{y \in \boldsymbol{Y}} \|x - y\|_2^2 + \frac{1}{2|\boldsymbol{Y}|} \sum_{y \in \boldsymbol{Y}} \min_{x \in \boldsymbol{X}} \|x - y\|_2^2. \end{aligned} \tag{14}$$

We sum over parts to obtain a total loss of

$$\mathcal{L}_{\boldsymbol{Y}} = \sum_{m=1}^{M} \mathcal{L}_c(\hat{\boldsymbol{X}}_m, \hat{\boldsymbol{Y}}_m), \tag{15}$$

Class	Model	JSD ↓	MMD-CD ↓	MMD-EMD ↓	COV-CD ↑	COV-EMD ↑
Chair	r-GAN (dense)⋆	0.238	0.0029	0.136	33	13
	r-GAN (conv)⋆	0.517	0.0030	0.223	23	4
	Valsesia (no up.)⋆	0.119	0.0033	0.104	26	20
	Valsesia (up.)⋆	0.100	0.0029	0.097	30	26
	TREEGAN (Shu, Park, and Kwon 2019)	0.119	0.0016	0.101	58	30
	MRGAN (Gal et al. 2020)	0.246	0.0021	0.166	**67**	23
	EDITVAE (M=7)	**0.063**	**0.0014**	**0.082**	46	**32**
	EDITVAE (M=3)	**0.031**	0.0017	0.101	45	**39**
Airplane	r-GAN(dense)⋆	0.182	0.0009	0.094	31	9
	r-GAN(conv)⋆	0.350	0.0008	0.101	26	7
	Valsesia (no up.)⋆	0.164	0.0010	0.102	24	13
	Valsesia (up.)⋆	0.083	0.0008	0.071	31	14
	TREEGAN (Shu, Park, and Kwon 2019)	0.097	**0.0004**	0.068	61	20
	MRGAN (Gal et al. 2020)	0.243	0.0006	0.114	**75**	21
	EDITVAE (M=6)	**0.043**	**0.0004**	**0.024**	39	**30**
	EDITVAE (M=3)	**0.044**	0.0005	**0.067**	23	17
Table	TREEGAN (Shu, Park, and Kwon 2019)	0.077	0.0018	0.082	71	**48**
	MRGAN (Gal et al. 2020)	0.287	0.0020	0.155	**78**	31
	EDITVAE (M=5)	0.081	**0.0016**	**0.071**	42	27
	EDITVAE (M=3)	**0.042**	**0.0017**	0.130	39	30

Table 1: Generative performance. ↑ means the higher the better, ↓ means the lower the better. The score is highlighted in bold if it is the best one compared with state-of-the-art. Here M is the number of minimum parts we expect to separate in training. For network with $\star$ we use the result reported in (Valsesia, Fracastoro, and Magli 2018; Shu, Park, and Kwon 2019)

where $\boldsymbol{X}_m$ is the subset of $\boldsymbol{X}$ whose nearest superquadric is $\boldsymbol{P}_m$, and $\hat{\boldsymbol{X}}_m = \mathcal{T}^{-1}(\boldsymbol{X}_m)$ is in canonical pose.

Superquadric Losses. The remaining terms in our loss relate to the part $\boldsymbol{P}_m$ and combined $\boldsymbol{P} = \bigcup_{m=1}^{M} \boldsymbol{P}_m$ primitives, and would match Paschalidou, Ulusoy, and Geiger (2019) but for the addition of a regulariser which discourages overlapping superquadrics, *i.e.*[1]

$$\mathcal{L}_o(\boldsymbol{P}) \\ = \frac{1}{M}\sum_{m=1}^{M} \frac{1}{|\boldsymbol{S}| - |\boldsymbol{S_m}|} \sum_{\boldsymbol{s}\in\boldsymbol{S}\setminus\boldsymbol{S}_m} \max\left(1 - H_m(\boldsymbol{s}), 0\right), \tag{16}$$

where $|\cdot|$ denotes cardinality, $\boldsymbol{S}_m$ is a point cloud sampled from $\boldsymbol{P}_m$, $\boldsymbol{S} = \bigcup_{m=1}^{M} \boldsymbol{S}_m$, and $H_m(\cdot)$ is the smoothed indicator function for $\boldsymbol{P}_m$ defined in Solina and Bajcsy (1990).

Architecture Details

EDITVAE framework is shown in Figure 2. The posterior $Q_\phi(\boldsymbol{z}|\boldsymbol{X})$ is based on the POINTNET architecture (Qi et al. 2017), with the same structure as Achlioptas et al. (2018). For $P_\theta(\boldsymbol{\zeta}|\boldsymbol{z})$, we apply the linear transform and partitioning of (11) for disentangled part representations followed by further shape and pose disentanglement. We use the generator of TREEGAN (Shu, Park, and Kwon 2019) as the decoder, modelling $g_{\boldsymbol{Y}}^{(i)}$, to generate the point cloud for each part. The super-quadric decoder modules match Paschalidou, Ulusoy, and Geiger (2019) for primitive generation $\boldsymbol{P}_m$, as do those for the $\mathcal{T}_m$. Weights are not shared among branches.

[1] $\mathcal{L}_o(\boldsymbol{P})$ matches the implementation of Paschalidou, Ulusoy, and Geiger (2019) provided by the authors.

Experiments

Evaluation metrics. We evaluate our EDITVAE on the ShapeNet (Chang et al. 2015) with the same data split as Shu, Park, and Kwon (2019) and report results on the three dominant categories of chair, airplane, and table. We adopt the evaluation metrics of Achlioptas et al. (2018), including Jensen-Shannon Divergence (JSD), Minimum Matching Distance (MMD), and Coverage (COV). As MMD and COV may be computed with either Chamfer Distance (CD) or Earth-Mover Distance (EMD), we obtain five different evaluation metrics, *i.e.* JSD, MMD-CD, MMD-EMD, COV-CD, and COV-EMD.

Baselines. We compare with four existing models of r-GAN (Achlioptas et al. 2018), Valsesia (Valsesia, Fracastoro, and Magli 2018), TREEGAN (Shu, Park, and Kwon 2019) and MRGAN (Gal et al. 2020). r-GAN and Valsesia generate point clouds as a single whole without parts inference or generation based on a tree structure as in TREEGAN. Similar to our approach, MRGAN performs unsupervised parts-aware generation, but with "parts" that lack a familiar semantic meaning and without disentangling pose.

Implementation details.[2] The input point cloud consists of a set of 2048 points, which matches the above baselines. Our prior on the global latent representation $\boldsymbol{z} \in \mathbb{R}^{256}$ is the usual standard Gaussian distribution. We chose $\boldsymbol{z}_{\boldsymbol{Y}}^{(m)} \in \mathbb{R}^{32}$, and $\boldsymbol{z}_{\mathcal{T}}^{(m)}$, $\boldsymbol{z}_{\boldsymbol{P}}^{(m)} \in \mathbb{R}^{8}$ for the local latents of (12). We trained EDITVAE using the ADAM optimizer (Kingma and Ba 2015) with a learning rate of 0.0001 for 1000 epochs and a batch size of 30. To fine-tune our model we adopted the β-VAE framework (Higgins et al. 2016).

[2] Code will be provided on publication of the paper.

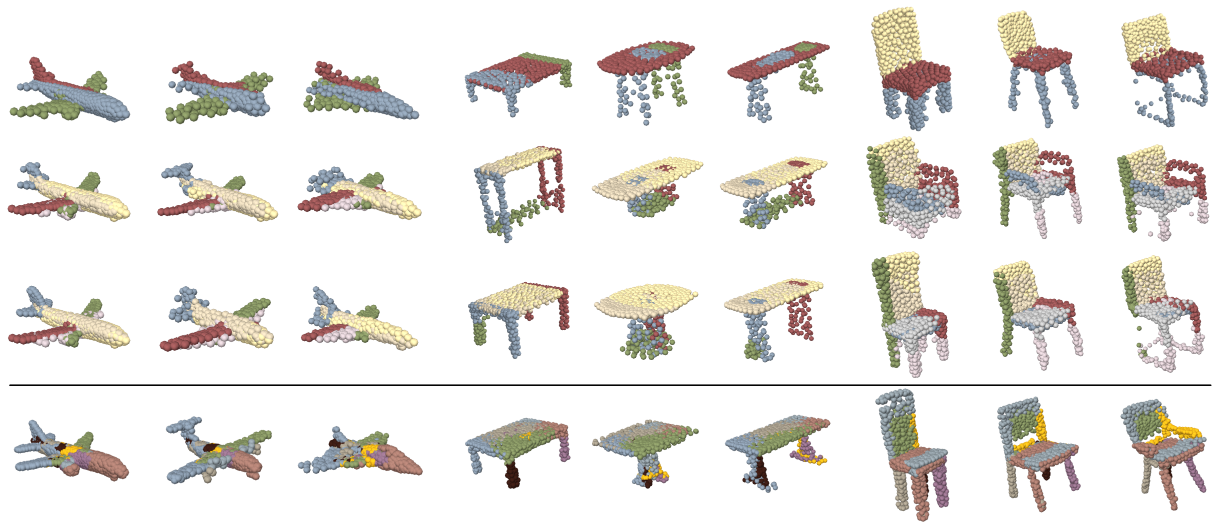

Figure 3: Parts-based generated point clouds from the airplane, table and chair categories, coloured by part. *Bottom row:* examples generated by TREEGAN (Shu, Park, and Kwon 2019). The top three rows are EDITVAE—the top row with $M = 3$, and the second and third rows with the number of parts M reported in Table 1.

Results

Generation. EDITVAE generates point clouds by simply sampling from a standard Gaussian prior for $\boldsymbol{z}$, mapping by A and and the subsequent part branches of Figure 2, before merging to form the complete point cloud. We show quantitative and qualitative results in Table 1 and Figure 3, respectively. As shown in Table 1, the proposed EDITVAE achieves competitive results (see *e.g.* the $M = 7$ results for the chair category) compared with the states of the art. The parts number M is manually selected to achieve a meaningful semantic segmentation, *e.g.* a chair may be roughly decomposed into back, base, and legs for $M = 3$. Furthermore, while Shu, Park, and Kwon (2019) generates point clouds according to a tree structure—and could thereby potentially generate points with consistent part semantics—it does not allow the semantics-aware shape editing due to lacking of disentangled parts representations. To the best of our knowledge, MRGAN (Gal et al. 2020) is the only other method achieving parts-disentangled shape representation and generation in an unsupervised manner. The results in Table 1 show that our method outperforms MRGAN in both the JSD and MMD metrics. Morover, EDITVAE achieves highly semantically meaningful parts generation as shown in Figure 3 and the experiment as discussed below, which further achieves parts-aware point cloud editing.

Parts Editing. EDITVAE disentangles the point clouds into latents for each part, and then in turn into the point cloud, pose, and primitive for each part. This design choice allows editing some parts with other parts fixed, yielding controllable parts editing and generation. We demonstrate this via both parts mixing and parts (re-)sampling.

Parts Mixing. It is defined by exchanging some parts between generated reference and ground-truth point clouds while keeping others fixed. We achieve mixing by transferring corresponding parts latents from reference to ground-truth, and further transforming it by the generator and pose of the parts in the ground-truth. The corresponding part in the ground-truth point cloud may therefore be changed to the style of the reference one. For example, the results in the first row of Figure 4 show that the ground-truth shape of a sofa with solid armed base may be changed into a larger hollow armed one based on its reference shape with consistent style. Namely, the size and pose of mixed parts follow that of the ground-truth, but keep the style from the reference.

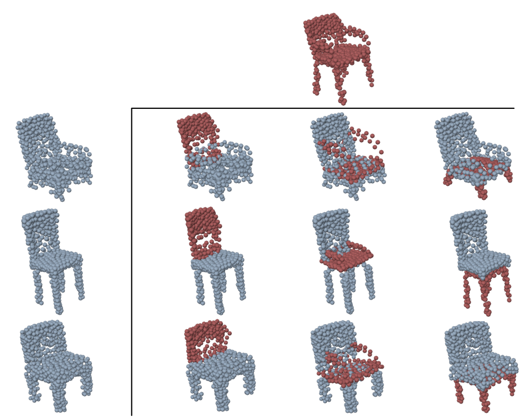

Figure 4: Parts mixing in the chair category with $M = 3$. *Far left:* ground truth point clouds, *top:* reference point cloud. *Remaining:* from left to right, back, base, and legs for ground truth points are mixed by corresponding parts in the reference one via mixing their disentangled latents.

Parts Sampling. This involves resampling some parts in a generated point cloud. For resampled parts, we fix the pose but resample the point cloud parts latent. The fixed pose is essential to maintain generated part point clouds with a con-

Model	Chair			Airplane			Table		
	TreeGAN	M=3	M=7	TreeGAN	M=3	M=6	TreeGAN	M=3	M=5
MCD↓	0.0164	0.0028	0.0121	0.0043	0.0016	0.0018	0.0266	0.0121	0.0214

Table 2: Semantic meaningfulness measurements. $M = *$ represents EditVAE inTable 1. The lower MCD the better.

Model	MMD-CD↓			
	as whole	base	back	leg
EditVAE	0.0017	0.0016	0.0014	0.0024
Baseline	0.0025	0.0017	0.0015	0.0024

Table 3: Generative performance for the entire shape and its parts, for the chair category. Semantic labels are obtained by primitive segmentation in our framework.

Model	JSD↓	MMD-CD↓	COV-CD↑
Baseline-G	0.062	0.0019	42
Baseline-S	0.163	0.0030	10
EditVAE (M=3)	0.031	0.0017	45
EditVAE (M=7)	0.063	0.0014	46

Table 4: Generative performance comparsion for EditVAE and two baselines in chair category.

sistent location that matches the other fixed parts to achieve controllable generation. Qualitative results for parts sampling are in Figure 5. Variations in the part styles demonstrated the controllable point cloud generation.

Semantic Meaningfulness. We first define a vanilla measurement by comparing the distance between the ground truth semantic label and the unsupervisedly generated one. The distance is defined as the mean of smallest Chamfer distance for each unsupervised part with respect to all ground truth parts (MCD in Table 2). As MRGAN (Gal et al. 2020) lacks accompanying code, we mainly compare the semantic meaningfulness with respect to TreeGAN in Table 2. EditVAE outperforms when we define the ground truth segmentation as the most meaningful.

Ablation Studies

Generation / Editing Trade-Off. We aim to evaluate the influence of the linear mapping A for disentangled representation learning (see Figure 2). To this end, we introduce a Baseline framework by simply removing this A. Results are shown in Table 3. Specifically, we compare our generation with the Baseline results at the whole point cloud level and at the parts level, such as the *base*, *leg*, and *back*, for the chair category. While Baseline achieves disentangled parts-aware representation learning and comparable results for parts sampling to EditVAE[3], the manner in which Baseline generates points as a whole via sampling from a standard Gaussian yields inferior performance due to the

[3] We evaluate each part generation result separately.

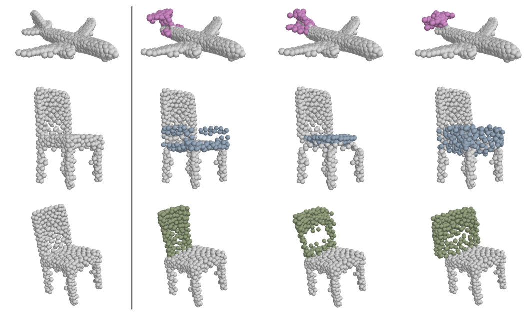

Figure 5: Parts sampling. *Far left:* the reference point clouds. Colored parts in the three right columns are sampled from latent space—from top to bottom, we sampled the airplane stabilizer, chair base, and chair back.

mismatched style across parts. Thus, the mapping A manages to decouple the undesirable generation / editing trade-off caused by disentanglement. Detailed analysis and visualizations are in the supplementary materials.

Stage-wise Baselines. We compared EditVAE with two stage-wise baselines defined as *Baseline-S* and *Baseline-G*. In particular, *Baseline-S* is built by first generating parts labels via the state-of-the-art unsupervised segmentation method (Paschalidou, Ulusoy, and Geiger 2019) followed by a supervised parts-aware generation approach (Schor et al. 2019). *Baseline-G* is created by training the the point cloud branch in Figure 2 with the ground-truth parts segmentation. The comparison is performed on the chair category in ShapeNet (Chang et al. 2015), and reported in Table 4.

EditVAE is robust to semantic segmentation as its generation is close to *Baseline-G*. Further, the performance of $M = 3$ is closer to *Baseline-G* compared with $M = 7$, in line with our observation (see Figure 3) that this case achieves a similar segmentation to the ground-truth. Further, EditVAE outperforms *Baseline-S* by overcoming the style-mismatch issue and is robust to noise introduced by mapping parts to a canonical system with learned poses.

Conclusions

We introduced EditVAE, which generates parts-based point clouds in an unsupervised manner. The proposed framework learns a disentangled latent representation with a natural inductive bias that we introduce by jointly modeling latent part- and pose-models, thereby making parts controllable. Through various experiments, we demonstrated that EditVAE balances parts-based generation and editing in a useful way, while performing strongly on standard point-cloud generation metrics.

Acknowledgements

Miaomiao Liu was supported in part by the Australia Research Council DECRA Fellowship (DE180100628) and ARC Discovery Grant (DP200102274).

References

Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; and Guibas, L. 2018. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, 40–49. PMLR.

Barr, A. H. 1987. Global and local deformations of solid primitives. In *Readings in Computer Vision*, 661–670. Elsevier.

Bongsoo Choy, C.; Stark, M.; Corbett-Davies, S.; and Savarese, S. 2015. Enriching object detection with 2d-3d registration and continuous viewpoint estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2512–2520.

Cai, R.; Yang, G.; Averbuch-Elor, H.; Hao, Z.; Belongie, S.; Snavely, N.; and Hariharan, B. 2020. Learning gradient fields for shape generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 364–381. Springer.

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

Chen, R. T. Q.; Li, X.; Grosse, R.; and Duvenaud, D. 2018. Isolating Sources of Disentanglement in Variational Autoencoders. *Advances in Neural Information Processing Systems*.

Chen, Z.; Yin, K.; Fisher, M.; Chaudhuri, S.; and Zhang, H. 2019. BAE-NET: branched autoencoder for shape co-segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8490–8499.

Dubrovina, A.; Xia, F.; Achlioptas, P.; Shalah, M.; Groscot, R.; and Guibas, L. J. 2019. Composite shape modeling via latent space factorization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8140–8149.

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

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2016. beta-vae: Learning basic visual concepts with a constrained variational framework.

Hui, L.; Xu, R.; Xie, J.; Qian, J.; and Yang, J. 2020. Progressive point cloud deconvolution generation network. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, 397–413. Springer.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Kalatzis, D.; Eklund, D.; Arvanitidis, G.; and Hauberg, S. 2020. Variational Autoencoders with Riemannian Brownian Motion Priors. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 5053–5066. PMLR.

Kim, H.; and Mnih, A. 2018. Disentangling by factorising. In *International Conference on Machine Learning*, 2649–2658. PMLR.

Kim, J.; Oh, T.-H.; Lee, S.; Pan, F.; and Kweon, I. S. 2019a. Variational Prototyping-Encoder: One-Shot Learning with Prototypical Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9462–9470.

Kim, M.; Wang, Y.; Sahu, P.; and Pavlovic, V. 2019b. Bayes-factor-vae: Hierarchical bayesian deep auto-encoder models for factor disentanglement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2979–2987.

Kim, M.; Wang, Y.; Sahu, P.; and Pavlovic, V. 2019c. Relevance factor vae: Learning and identifying disentangled factors. *arXiv preprint arXiv:1902.01568*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Li, R.; Li, X.; Hui, K.-H.; and Fu, C.-W. 2021. SP-GAN: Sphere-guided 3D shape generation and manipulation. *ACM Transactions on Graphics (TOG)*, 40(4): 1–12.

Locatello, F.; Abbati, G.; Rainforth, T.; Bauer, S.; Schölkopf, B.; and Bachem, O. 2019a. On the Fairness of Disentangled Representations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 14584–14597.

Locatello, F.; Bauer, S.; Lucic, M.; Raetsch, G.; Gelly, S.; Schölkopf, B.; and Bachem, O. 2019b. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, 4114–4124. PMLR.

Locatello, F.; Tschannen, M.; Bauer, S.; Rätsch, G.; Schölkopf, B.; and Bachem, O. 2020. Disentangling Factors of Variations Using Few Labels. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mo, K.; Guerrero, P.; Yi, L.; Su, H.; Wonka, P.; Mitra, N. J.; and Guibas, L. J. 2019. StructureNet: hierarchical graph networks for 3D shape generation. *ACM Trans. Graph.*, 38(6): 242:1–242:19.

Mo, K.; Wang, H.; Yan, X.; and Guibas, L. 2020. PT2PC: Learning to generate 3d point cloud shapes from part tree conditions. In *European Conference on Computer Vision*, 683–701. Springer.

Nash, C.; and Williams, C. K. 2017. The shape variational autoencoder: A deep generative model of part-segmented 3D objects. In *Computer Graphics Forum*, volume 36, 1–12. Wiley Online Library.

Nielsen, D.; Jaini, P.; Hoogeboom, E.; Winther, O.; and Welling, M. 2020. SurVAE Flows: Surjections to Bridge the Gap between VAEs and Flows. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Paschalidou, D.; Ulusoy, A. O.; and Geiger, A. 2019. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10344–10353.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Rezende, D.; and Mohamed, S. 2015. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 1530–1538. PMLR.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In Xing, E. P.; and Jebara, T., eds., *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, 1278–1286. Bejing, China: PMLR.

Schor, N.; Katzir, O.; Zhang, H.; and Cohen-Or, D. 2019. Componet: Learning to generate the unseen by part synthesis and composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8759–8768.

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

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28: 3483–3491.

Solina, F.; and Bajcsy, R. 1990. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE transactions on pattern analysis and machine intelligence*, 12(2): 131–147.

Sun, C.-Y.; Zou, Q.-F.; Tong, X.; and Liu, Y. 2019. Learning adaptive hierarchical cuboid abstractions of 3d shape collections. *ACM Transactions on Graphics (TOG)*, 38(6): 1–13.

Tulsiani, S.; Su, H.; Guibas, L. J.; Efros, A. A.; and Malik, J. 2017. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2635–2643.

Valsesia, D.; Fracastoro, G.; and Magli, E. 2018. Learning localized generative models for 3d point clouds via graph convolution. In *International conference on learning representations*.

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

Yang, K.; and Chen, X. 2021. Unsupervised Learning for Cuboid Shape Abstraction via Joint Segmentation from Point Clouds. *arXiv preprint arXiv:2106.03437*.

Yang, X.; Wu, Y.; Zhang, K.; and Jin, C. 2021. CPCGAN: A Controllable 3D Point Cloud Generative Adversarial Network with Semantic Label Generating. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3154–3162.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.