Point Cloud Part Editing: Segmentation, Generation, Assembly, and Selection

Kaiyi Zhang[1], **Yang Chen**[1], **Ximing Yang**[1], **Weizhong Zhang**[1,2], **Cheng Jin**[1,2]

[1]School of Computer Science, Fudan University, Shanghai, China
[2]Innovation Center of Calligraphy and Painting Creation Technology, MCT, China
{zhangky20, chen_yang19, xmyang19, weizhongzhang, jc}@fudan.edu.cn

Abstract

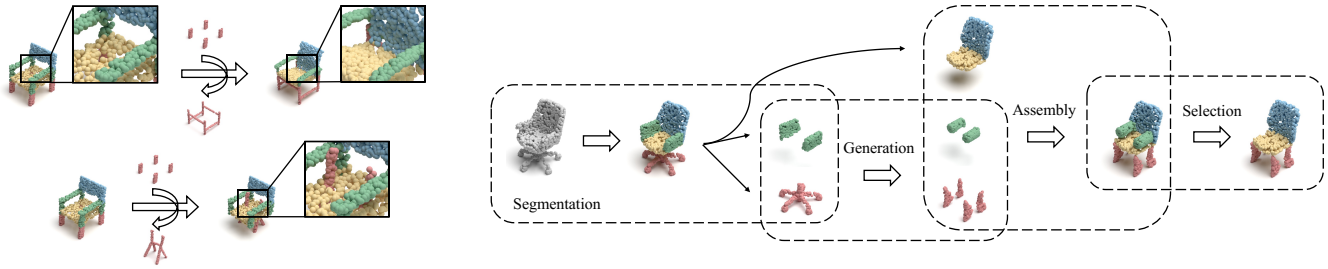
Ideal part editing should guarantee the diversity of edited parts, the fidelity to the remaining parts, and the quality of the results. However, previous methods do not disentangle each part completely, which means the edited parts will affect the others, resulting in poor diversity and fidelity. In addition, some methods lack constraints between parts, which need manual selections of edited results to ensure quality. Therefore, we propose a four-stage process for point cloud part editing: **S**egmentation, **G**eneration, **A**ssembly, and **S**election. Based on this process, we introduce **SGAS**, a model for part editing that employs two strategies: feature disentanglement and constraint. By independently fitting part-level feature distributions, we realize the feature disentanglement. By explicitly modeling the transformation from object-level distribution to part-level distributions, we realize the feature constraint. Considerable experiments on different datasets demonstrate the efficiency and effectiveness of SGAS on point cloud part editing. In addition, SGAS can be pruned to realize unsupervised part-aware point cloud generation and achieves state-of-the-art results.

Introduction

In the context of 3D object modeling, parts are considered the fundamental units. Recently, part-based methods (Wang et al. 2018; Mo et al. 2019a; Li, Niu, and Xu 2020; Jones et al. 2020; Gal et al. 2021; Li, Liu, and Walder 2022) have become more and more prevailing. These methods typically involve obtaining different parts first and then assembling them. Although many works have explored procedural content generation (Liu et al. 2021), which is often used to make material maps and game maps, the rapid development of game scenes still relies heavily on the generation of 3D objects. Part-based methods enable part editing, which involves replacing some parts of an object to create a new one, thereby enhancing the diversity of 3D modeling.

Ideal part editing should make edited parts diverse while keeping unedited parts unchanged to form a reasonable object. These correspond to three important properties of the edited results: diversity, fidelity, and quality. However, when previous part-based methods are applied to part editing, they have two problems. As shown in Figure 1(a), on the one hand, methods such as MRGAN (Gal et al. 2021) and SP-GAN (Li et al. 2021) do not realize radical disentanglement between parts. Therefore, when some parts are modified, other parts will also change, which means they do not guarantee the fidelity to the remaining parts. Similarly, multi-modal shape completion methods such as MSC-cGAN (Wu et al. 2020a), which can be regarded as a subset of part editing, also do not disentangle parts. This not only changes the input parts but also makes the completion results less diverse. Someone may argue that the adjacent parts may change to accommodate the edited parts, but this will not affect the requirement of radical disentanglement, since all changed parts can be regarded as edited parts. On the other hand, some methods (Schor et al. 2019; Li, Niu, and Xu 2020; Li, Liu, and Walder 2022) do not implement constraints between parts effectively. This may result in poor part assembly and mismatched parts to be used in the formation of an object. Although these methods attempt to achieve assembly by moving parts, the changed parts still need a manual selection to ensure high-quality edited results that lead to a reasonable object.

To address these issues, as shown in Figure 1(b), we first propose a four-stage process for point cloud part editing: segmentation, generation, assembly, and selection. For the three properties of edited results, segmentation can guarantee the fidelity to the remaining parts by isolating the parts; generation can guarantee the diversity of edited parts by exploring different variations of the parts; assembly and selection can guarantee the quality of the results by choosing the most appropriate edited parts and assembling them in a coherent manner. Based on this process, we introduce a model SGAS for part editing that employs two strategies: feature disentanglement and constraint. We first use unsupervised shape co-segmentation methods (Chen et al. 2019; Zhu et al. 2020; Zhang et al. 2022) or manual segmentation to obtain Ground Truth parts. Then we pre-train several autoencoders at the part level. Finally, by adversarially supervising part-level feature transformations, we realize the feature disentanglement during generation. Since each part is generated separately, this not only ensures the diversity of edited parts but also the fidelity to the remaining parts. In addition, we make the distribution of each part transformed from the same Gaussian distribution and adversarially supervise the generations of all parts simultaneously to ensure that



(a) Two problems with existing part editing methods

(b) A four-stage point cloud part editing process

Figure 1: (a) Top: w/o disentanglement between parts. When changing the chair base, the other parts such as the chair back, arm, and seat will also change. Bottom: w/o constraints between parts. The changed chair base is not only poorly assembled but also does not match other parts, resulting in the generation of an unreasonable object. (b) It includes four stages: segmentation, generation, assembly, and selection, which can guarantee fidelity, diversity, and quality of the edited results respectively.

edited parts can assemble well and form a reasonable object. This strategy is called feature constraint. It guarantees the quality of the edited results. By adding a part selection module to the final output part features, which allows SGAS to autonomously choose which parts do not need to be output, the quality of the edited results can be further improved.

Our main contributions are the following:

- We propose a novel point cloud part editing process. It inlcudes four stages: segmentation, generation, assembly, and selection.
- Based on the proposed process, we introduce SGAS, a model for part editing that employs two strategies of feature disentanglement and constraint. Experiments show that SGAS achieves excellent quantitative and qualitative part editing results.
- A new diversity metric of edited results: Total Mutual Difference Surface (TMDS).
- SGAS can be pruned to realize unsupervised part-aware point cloud generation and achieves state-of-the-art results on the ShapeNet-Partseg dataset.

Related Work

Part-based Shape Generation

Unsupervised Part-aware Point Cloud Generation The fine-grained improvement of generative results can be achieved through local generation. Therefore, many methods attempt to explore the generation of multiple parts and combine them into a final shape. Since part-level ground truth data is often unavailable, these methods typically involve unsupervised segmentation of parts. For example, TreeGAN (Shu, Park, and Kwon 2019) first designs the generation of the point cloud as a tree growth process and then combines the various parts at the leaf nodes. To achieve controllable point cloud generation, SP-GAN (Li et al. 2021) is proposed. Similar to FoldingNet (Yang et al. 2018), SP-GAN transforms a sphere in 3D space into a target point cloud, where different parts of the 3D sphere correspond to different parts of the target point cloud. MRGAN (Gal et al. 2021) explicitly realizes part disentanglement by using multiple branches of tree-structured graph convolution layers. Instead of supervising each part respectively, it conducts overall supervision after assembling all the parts. Considering that the parts of MRGAN lack semantic meaning, Li, Liu, and Walder (2022) propose EditVAE, which can achieve semantics-aware point cloud generation. Each branch of EditVAE generates not only parts but also additional part offsets and primitives for auxiliary supervision. In addition, (Öngün and Temizel 2020; Postels et al. 2021; Li and Baciu 2022; Cheng et al. 2022) also play important roles in promoting part-aware point cloud generation.

Assembly-based Shape Generation Many datasets, such as ShapeNet-Partseg (Yi et al. 2016) and PartNet (Mo et al. 2019b), provide part-level semantics. Therefore, many works explore shape generation by assembling parts. Specifically, these works can be roughly categorized into three groups: (1) assemble without generation (Schor et al. 2019; Dubrovina et al. 2019; Yin et al. 2020; Hui et al. 2022; Wu et al. 2023). These methods only reconstruct the parts and achieve the diversity of results by assembling different parts. For example, CompoNet (Schor et al. 2019) synthesizes ”unseen” but reasonable point clouds by varying both the parts and their compositions. Dubrovina et al. (2019) propose a semantic-part-aware embedding space to realize shape composition and decomposition. PartAttention (Wu et al. 2023) uses a part-wise attention framework to achieve affine transformation of the decoded parts. (2) assemble after generation (Li et al. 2017; Wang et al. 2018; Wu et al. 2019; Li, Niu, and Xu 2020). In contrast to the first category, the parts used for assembling are generated from a Gaussian distribution. These methods mainly focus on designing part offset networks to efficiently assemble parts. For example, G2LGAN (Wang et al. 2018) uses global and local GANs to supervise the correlation between the parts and the quality of each part respectively. It also adds a Part Refiner to optimize the generated results, such as removing outliers and completing missing regions. PAGENet (Li, Niu, and Xu 2020) generates parts using part-level VAEs and designs a Part Assembler to translate parts based on some anchor parts. (3) assemble while generating (Zou et al. 2017; Mo et al. 2019a, 2020; Wu et al. 2020b; Jones et al. 2020; Wang et al. 2022; Zhuang 2022). This type of method

does not assemble the parts after generating all of them, but rather assembles them progressively during generation. For example, 3D-PRNN (Zou et al. 2017) proposes a generative recurrent neural network that synthesizes multiple plausible shapes step-by-step based on primitives. This progressive process preserves long-range structural coherence. PQ-Net (Wu et al. 2020b) adopts RNN structure and learns 3D shape representations as a sequential part assembly. ShapeAssembly (Jones et al. 2020) achieves 3D shape structure synthesis by generating domain-specific language programs. The transformation of the statements in these programs enables ShapeAssembly to control the generated results.

Multimodal Shape Completion

Shapes with missing semantics can lead to a variety of completion results. For example, Wu et al. (2020a) propose MSC-cGAN. Based on pcl2pcl (Chen, Chen, and Mitra 2020), it adds an additional Gaussian distribution during the transformation from partial to complete point cloud features and an encoder to guarantee completion fidelity to the input partial. Different samples on Gaussian distribution correspond to different completion results. Zhou, Du, and Wu (2021) introduce PVD, a unified probabilistic formulation, to achieve multimodal shape completion by progressively removing noise from the samples. AutoSDF (Mittal et al. 2022) utilizes a transformer-based autoregressive model to generate patch embeddings extracted independently by VQ-VAE (van den Oord, Vinyals, and kavukcuoglu 2017) step-by-step. The idea of ShapeFormer (Yan et al. 2022) is similar to AutoSDF. The difference is that AutoSDF embeds the whole 3D space while ShapeFormer introduces a compact 3D representation VQDIF that embeds only the space occupied by 3D shapes, making it more efficient. There are also many other works (Arora et al. 2022; Zhang et al. 2021; Zhao et al. 2021; Jiang and Daniilidis 2022; Cheng et al. 2023) exploring multimodal shape completion.

Method

In this section, we first describe the architecture of SGAS according to the proposed four-stage point cloud part editing process, and then give the loss functions of SGAS.

Segmentation

To achieve the disentanglement between parts which can guarantee fidelity in part editing, SGAS is designed to have multiple branches. Each branch generates a part and requires a Ground Truth part for supervision. In our opinion, parts can be semantic parts, such as a chair back, seat, and base, and can also be local areas of a shape's surface. The former can directly use some datasets (Yi et al. 2016; Mo et al. 2019b) with part semantic labels to obtain Ground Truth parts, while the latter need some unsupervised shape co-segmentation methods (Chen et al. 2019; Zhu et al. 2020; Zhang et al. 2022) to obtain Ground Truth parts. We follow the idea of l-GAN (Achlioptas et al. 2017), which demonstrates that generating on features is better than directly generating on point clouds. Therefore, using the segmented Ground Truth parts, we pre-train an autoencoder for each semantic part to convert point clouds into features. The encoder is the same as PointNet (Qi et al. 2017) encoder. The decoder uses a fully connected network. Earth Mover's Distance(EMD) (Fan, Su, and Guibas 2017) is used to supervise the training of these autoencoders. As shown in Figure 2, the trained encoders $E_{pi}, i = 1...n$ and decoders $D_{pi}, i = 1...n$ are used to build SGAS. They do not update parameters during SGAS training.

Generation

The input of SGAS includes not only Gaussian noise but also unedited parts. The purpose is to realize that the style of generated parts matches that of unedited parts, thereby ensuring the quality of the final edited results. We use AdaIN Layer (Huang and Belongie 2017) to integrate the unedited parts into the Gaussian distribution. Specifically, a PointNet encoder E is used to encode the unedited parts to the mean μ and standard deviation Σ of a Gaussian distribution. The μ and Σ are then applied to the standard Gaussian distribution to obtain a new Gaussian distribution $\mathcal{N}(\mu, \Sigma^2)$. Based on this new distribution, we design several part-level GANs to generate parts. As shown in Figure 2, a Gaussian noise $z \in \mathbb{R}^{128} \sim \mathcal{N}(\mu, \Sigma^2)$ is transformed by generators $G_{pi}, i = 1...n$ into part latent features to realize feature disentanglement. The generator uses a 3-layer fully connected network $(256, 512, 128)$. The dimension of the part latent feature is 128. Part features are then sent to discriminators $F_{pi}, i = 1...n$ to distinguish real parts and generated parts. The discriminator uses a 3-layer fully connected network $(256, 512, 1)$. These part-level discriminators ensure the quality of each part.

Assembly

Since the branches used for part generation are independent, the generated parts may not be assembled into a reasonable object. To solve this problem, as shown in Figure 2, we add a global discriminator F in SGAS to supervise all generated parts simultaneously, which realizes feature constraint. The discriminator uses a 3-layer fully connected network $(256, 512, 1)$. Compared to methods (Schor et al. 2019; Dubrovina et al. 2019; Yin et al. 2020; Li, Niu, and Xu 2020) that use affine transformation to realize part assembly, our global discriminator can further ensure matching between the generated parts. Since some parts of the target point cloud already exist, we use Part Mask to replace some generated parts. Specifically, we first use pre-trained encoders $E_{pi}, i = 1...n$ to obtain the part latent features of unedited parts. These part latent features are then used to replace the corresponding generated part features. Finally, the replaced features are sent to discriminator F.

Selection

In a real scene, an object does not necessarily contain all semantic parts. For example, a chair without arms and a lamp without a holder. In order to realize this, we perform Part Select on the part features processed after Part Mask. Part Select uses a threshold τ to filter the parts that SGAS thinks do not need to be output. It does not need training.

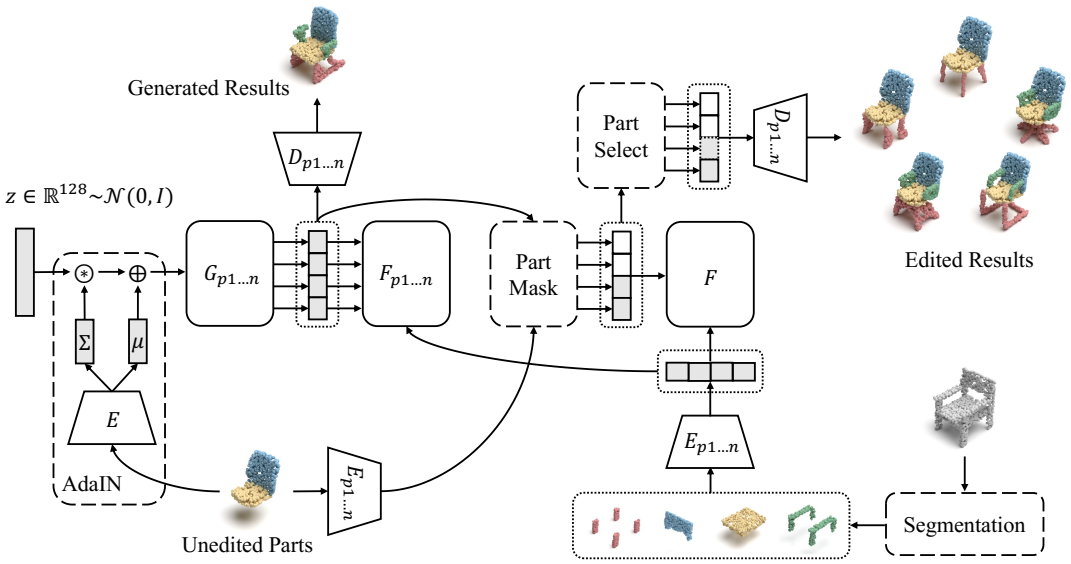


Figure 2: The architecture of SGAS. The inputs are Gaussian noise and unedited parts. The outputs are diverse generated or edited results. SGAS obtains Ground Truth parts through segmentation and uses them to pre-train part-level autoencoders, which convert point clouds into features. By incorporating the style of unedited parts into the Gaussian distribution using an AdaIN layer and performing part-level GAN supervision, SGAS can generate new parts. To constrain each part to form a reasonable object, SGAS applies part masking and uses a global discriminator. Finally, SGAS performs part selection on each part feature, allowing the model to autonomously choose which parts do not need to be output.

Specifically, since some parts might not exist in the Ground Truth point clouds, we set the latent features corresponding to these parts to zero. Therefore, the trained SGAS can automatically determine whether a part needs to be output. It forces the features of parts that do not need to be output as close to zero as possible. In this way, By not decoding the parts whose features are within the threshold τ, we realize part selection in the output point clouds. The filter conditions for Part Select are given as:

$$\left| \frac{1}{n} \sum_{i=1}^{n} G_{pi}(z) \right| \leq \tau \tag{1}$$

where $|\cdot|$ represents absolute value, n is the number of parts.

Loss Functions

We adopt the loss function introduced in Wasserstein GAN (Arjovsky, Chintala, and Bottou 2017) with gradient penalty (Gulrajani et al. 2017). Network E, $G_{pi}, i = 1...n$, $F_{pi}, i = 1...n$, and F need training. The losses are given as:

$$\begin{aligned} \mathcal{L}_G = &-\alpha * \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{z \sim Z}[F_{pi}(G_{pi}(z))] \\ &-\beta * \mathbb{E}_{z \sim Z}[F(\bigcup_{i=1}^{n} G_{pi}(z))] \end{aligned} \tag{2}$$

$$\begin{aligned} \mathcal{L}_{F_p} = &\frac{1}{n} \sum_{i=1}^{n} (\mathbb{E}_{z \sim Z}[F_{pi}(G_{pi}(z))] - \mathbb{E}_{x_i \sim R_{xi}}[F_{pi}(x_i)] \\ &+\lambda_{gp} \mathbb{E}_{\hat{x_i}}[(\|\nabla_{\hat{x_i}} F_{pi}(\hat{x_i})\|_2 - 1)^2]) \end{aligned} \tag{3}$$

$$\begin{aligned} \mathcal{L}_F = &\mathbb{E}_{z \sim Z, x_i \sim P_{xi}}[F(\bigcup_{i=1}^{n} M_i G_{pi}(z)) + (1 - M_i) E_{pi}(x_i)] \\ &-\mathbb{E}_{x_i \sim R_{xi}}[F(\bigcup_{i=1}^{n} x_i)] + \lambda_{gp} \mathbb{E}_{\hat{x_i}}[(\|\nabla_{\hat{x_i}} F(\bigcup_{i=1}^{n} \hat{x_i})\|_2 - 1)^2] \end{aligned} \tag{4}$$

where $\mathcal{L}_G$, $\mathcal{L}_{F_p}$, and $\mathcal{L}_F$ represent the loss functions of the part-level generator, the part-level discriminator, and the global discriminator respectively. α and β are hyperparameters that control the proportion of the part-level to the global GAN. N is the number of parts. $x_i, i = 1...n$ are parts. The formulas after λ_{gp} are the gradient penalty terms proposed by Gulrajani et al. (2017). $Z = \mathcal{N}(\mu, \Sigma^2)$, where μ and Σ are from encoder E. M is determined by the unedited parts.

Experiments

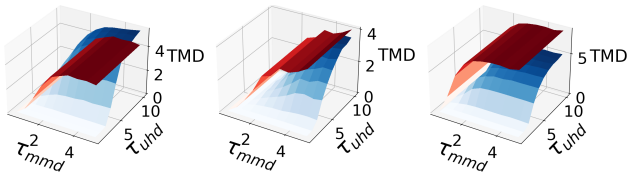
Datasets and Implementation Details

We evaluate SGAS on PartNet (Mo et al. 2019b) dataset. By merging fine-grained semantic labels and removing some special objects, we create a new dataset called PartNet.v0.Merged for point cloud part editing. Following previous works (Gal et al. 2021; Li, Liu, and Walder 2022), we perform unsupervised part-aware point cloud generation on ShapeNet-Partseg (Yi et al. 2016) dataset. We do not use semantic labels in the ShapeNet-Partseg dataset.

Adam optimizers are used for SGAS with a learning rate of $\alpha = 0.0005$, coefficients $\beta_1 = 0.5$ and $\beta_2 = 0.99$. All the experiments are performed on a single NVIDIA TITAN Xp for 2000 epochs with a batch size of 200. In loss functions,

	Model	Chair	Lamp	Table	Average
MMD ↓	MSC-cGAN	1.62	3.41	1.39	2.14
	SGAS	**1.33**	**2.26**	**1.06**	**1.55**
TMD ↑	MSC-cGAN	**5.45**	3.94	5.14	4.84
	SGAS	4.36	**4.48**	**8.04**	**5.63**

Table 1: Diversity part editing performance. MMD and TMD measure the quality and diversity respectively.



(a) Chair (b) Lamp (c) Table

Figure 3: Performance on the new metric Total Mutual Difference Surface (TMDS). Red represents SGAS; blue represents MSC-cGAN. The smaller the thresholds of MMD and UHD are, the more referential the calculated TMD is.

α and β are set to 1 and 1. λ_{gp} is set to 10. The threshold in Part Select is set to 0.5. We update the discriminator 5 times for each update of the generator. Each shape has 2048 points while each part has $\lfloor \frac{2048}{n} \rfloor$ points. n is the number of parts. During training, the input unedited parts for SGAS are obtained by randomly removing 1 to $n-1$ parts of objects in PartNet.v0.Merged dataset.

Point Cloud Part Editing

Diversity part editing is a commonly used operation in point cloud part editing that involves generating some parts multiple times to obtain various results. Previous part editing methods (Li, Niu, and Xu 2020; Gal et al. 2021; Li et al. 2021; Li, Liu, and Walder 2022) lack related evaluation metrics. Since diversity part editing has some intersects with multimodal shape completion, here we use the metrics MMD, TMD, and UHD adopted by Wu et al. (2020a) to measure the quality, diversity, and fidelity of the edited results respectively. Our SGAS disentangles each part, so the input unedited parts can remain unchanged. Therefore, we only compare MMD and TMD. Considering previous part editing methods can only perform part editing on generated objects, which makes it impossible to obtain Gaussian noise corresponding to existing objects for editing. Hence, we use the representative multimodal shape completion method MSC-cGAN (Wu et al. 2020a) as the baseline for the this study. As shown in Table 1, SGAS achieves excellent results in three representative categories.

During experiments, we find that the diversity metric TMD only measures the difference between edited results without considering fidelity and quality, which means two incorrect situations that can also result in high TMD: (a) the change of input unedited parts (corresponds to large UHD); (b) the edited parts with large differences but poor quality (corresponds to large MMD). Therefore, we further propose

$\alpha:\beta$	1:10	1:2	1:1	2:1	5:1	10:1
MMD ↓	1.47	1.39	**1.33**	1.35	1.36	1.40
TMD ↑	1.89	2.65	4.36	4.75	5.34	**7.68**

Table 2: Ablation results for the hyperparameters α and β. The set of $\alpha:\beta$ can be determined by requirement.

a new metric TMDS (TMD Surface) to solve the problems. Each point value on the surface is calculated as:

$$\mathrm{TMDS}(\tau_{\mathrm{uhd}}, \tau_{\mathrm{mmd}}) = \underset{p\in\mathbb{P}}{mean} \begin{cases} \mathrm{TMD}(s_1,..,s_k), \\ \quad \text{if } \exists s_i, i=1...k, \\ \quad\quad \mathrm{UHD}(p,s_i) \le \tau_{\mathrm{uhd}} \\ \quad\quad \mathrm{MMD}(s_i,\mathbb{D}) \le \tau_{\mathrm{mmd}} \\ 0, \quad \text{otherwise} \end{cases}$$

$$\mathrm{TMD}(s_1,..,s_k) = \sum_{i=1}^{k} \frac{1}{k-1} \sum_{j\neq i, j=1}^{k} \mathrm{CD}(s_i, s_j) \tag{5}$$

where p is input unedited parts, $s_i, i=1...k$ are K (e.g. 10) edited results. $\mathbb{P}$ is the unedited parts test dataset and $\mathbb{D}$ is orignal test dataset. CD is Chamfer Distance (Fan, Su, and Guibas 2017). τ_{uhd} and τ_{mmd} are thresholds of UHD and MMD respectively. TMDS requires that each point value on the surface is calculated when K edited results are guaranteed to satisfy the corresponding UHD and MMD thresholds. The smaller the thresholds of MMD and UHD are, the more referential the calculated TMD is. Therefore, as shown in Figure 3, we can find that the editing diversity of SGAS is better than that of MSC-cGAN in the Chair category.

We use SGAS to perform various part editing operations on PartNet.v0.Merged dataset. Figure 4(a) is the visualized comparison of the diversity edited results. We also perform our SGAS on three new categories: Display, Knife, and Mug. It can be found that SGAS can not only keep the input unchanged but also have higher editing diversity and quality. Figure 4(b) is a multi-round editing workflow achieved by SGAS. It demonstrates SGAS's ability to re-edit unsatisfactory edited results. Through three rounds of re-editing, a chair with right-angle arms and wheels is edited into a chair with circular arms and a circular base. Figure 4(c) shows some interpolation part editings. If the chairs in the left and right box are generated by Gaussian noise z_s and z_t respectively, the chairs between boxes are generated by Gaussian noise $z = (1-\alpha)z_s + \alpha z_t$. α increases from 0 to 1. From top to bottom, each row represents an interpolation of one, two, and three parts. We can clearly observe the gradual change process of the parts. Figure 4(d) demonstrates a style mixing part editing realized by SGAS. We first select four chairs with different styles. Then we edit different parts (colored) in different chairs. Finally, these edited parts are assembled to obtain chairs with styles from different chairs. This editing operation helps to create more diverse results.

As the hyperparameters α and β represent diversity and quality respectively, and have a significant impact on the edited results. Hence, we conduct an ablation study on them.

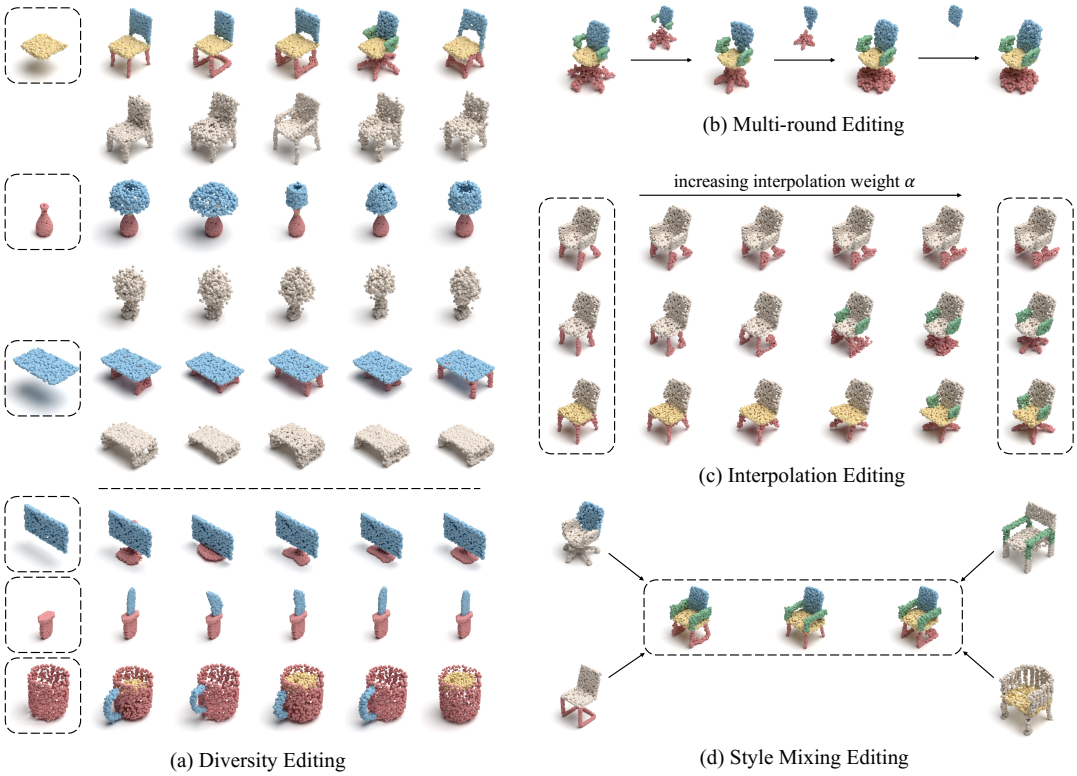


Figure 4: Various part editing operations. (a) Diversity editing comparison. The unedited parts are boxed, followed by five different edited results. The results of MSC-cGAN are uncolored while ours are colored by parts. (b) Re-editing of unsatisfactory edited results. The parts above the arrow are edited in each round. (c) Continuous transformation of selected parts (colored) through interpolation of two input Gaussian noises. Each row represents an interpolation of different number of parts. (d) Style Mixing of different parts in different objects. The mixed results are boxed.

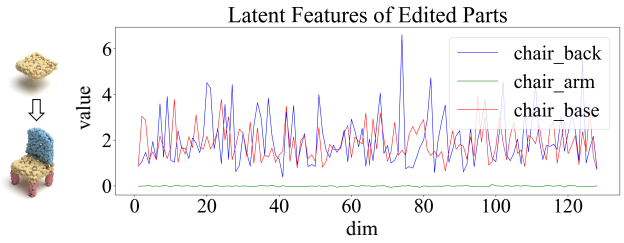


Figure 5: Edited parts with their corresponding latent features (128 dim). The chair without outputting arms has its corresponding latent feature of the chair arm near zero.

The results are presented in Table 2. It is observed that as $\alpha : \beta$ increase, the TMD also increase. However, the MMD, which measures the quality of the edited results, initially improves but then worsens. Therefore, we finally choose $\alpha : \beta = 1 : 1$ to realize part editing. However, if diversity is more important for the edited results, a higher $\alpha : \beta$ can also be used. To demonstrate SGAS's ability to automatically select parts, we visualize the latent features of the edited parts. As shown in Figure 5, it can be found that the latent feature of the chair arm is near zero, which means

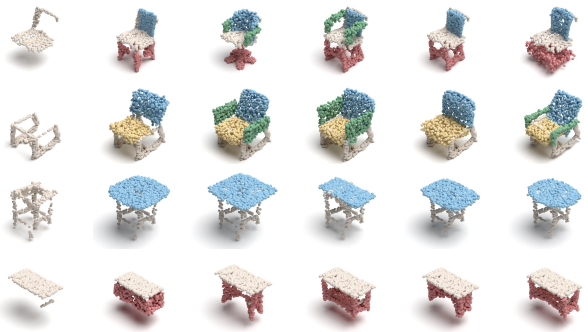


Figure 6: Performance on ScanNet. The leftmost column is incomplete input, followed by five diversity editing results.

SGAS believes that the newly generated chair arm is inappropriate. Therefore, the Part Select module in SGAS will filter this latent feature to prevent the chair arm from being output. To further prove the generalization ability of SGAS, we train SGAS on PartNet.v0.Merged dataset and test it on ScanNet (Dai et al. 2017) dataset. The results can be found in Figure 6. For parts with high missing rates, we will regenerate them, such as the chair back in the 1st row, and

Category	Model	JSD ↓	MMD ↓		COV %, ↑	
			CD	EMD	CD	EMD
Chair	TreeGAN	0.119	*0.0016*	0.101	58	30
	MRGAN	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*
	SGAS (N=7)	*0.047*	0.0020	**0.076**	*60*	**58**
Airplane	TreeGAN	0.097	0.0004	0.068	61	20
	MRGAN	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
	SGAS (N=6)	**0.036**	**0.0004**	*0.039*	*61*	**58**
Table	TreeGAN	0.077	0.0018	0.082	*71*	*48*
	MRGAN	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
	SGAS (N=5)	*0.057*	0.0020	**0.069**	65	**65**

Table 3: Generative performance. The optimal and suboptimal results are highlighted in bold and italics respectively. M and N represent the number of parts.

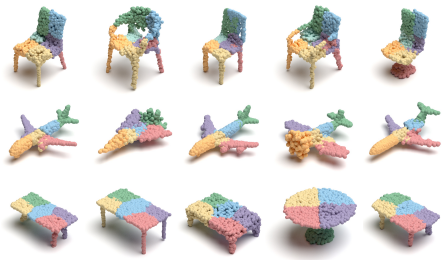


Figure 7: Point clouds generated by SGAS, colored by parts.

the newly generated parts are compatible with the existing incomplete parts. For parts with low missing rates, we will keep them directly. It can be found that even on unseen objects, SGAS's diversity editing results are still good.

Unsupervised Part-aware Point Cloud Generation

By pruning SGAS, including removing unedited parts input, AdaIN Layer, Part Mask, and Part Select, SGAS can be applied to realize unsupervised part-aware point cloud generation. It includes two steps: (a) modifying an unsupervised shape co-segmentation method AXform (Zhang et al. 2022) to get Ground Truth part datasets; (b) training SGAS on these part datasets. Specifically, we first modify the multi-branch AXform to output one structure point per branch. Second, we use these structure points to co-segment the Ground Truth point clouds into n part datasets. Third, the segmented parts are pre-encoded into latent features. Finally, these latent features are used to supervise the training of SGAS. However, we found that there might be some large gaps between parts during generation. Therefore, to achieve seamless generation, during unsupervised part segmentation, we further expand the number of points per part from $\lfloor \frac{2048}{n} \rfloor$ to $(1+\gamma)\lfloor \frac{2048}{n} \rfloor$. Here $\gamma = 0.1$. In the final output, we downsample the point cloud to 2048 points.

#Parts (n)	JSD ↓	MMD ↓		COV %,↑	
		CD	EMD	CD	EMD
2	0.042	0.0004	0.045	*61*	47
3	0.039	*0.0004*	0.043	60	52
5	0.040	0.0005	0.042	60	50
6	*0.036*	**0.0004**	**0.039**	**61**	**58**
8	0.040	0.0005	0.044	60	46
13	**0.035**	0.0005	*0.040*	60	*53*

Table 4: Ablation results for the number of parts. The optimal and suboptimal results are highlighted in bold and italics respectively. $n = 6$ is a suitable number of parts.

The quantitative results are shown in Table 3. The metrics are proposed by Achlioptas et al. (2017), and the results of previous methods are obtained from EditVAE (Li, Liu, and Walder 2022). MMD and COV represent the quality and diversity of the generated results respectively. It can be found that SGAS achieves excellent results overall, with the most number of top two metrics. Especially on COV-EMD, which represents diversity, SGAS has a significant improvement. Figure 7 gives visualized results of the generated point clouds. Different colors correspond to different parts. It intuitively illustrates the diversity and quality of the results generated by SGAS. We also conduct an ablation study on the number of parts in the Airplane category. As shown in Table 4, more or fewer parts are not necessarily beneficial to the results. Therefore, we chose the number of parts $N = 6$.

Conclusion

Previous methods do not disentangle each part completely or lack constraints between parts, which leads to poor diversity, fidelity, and quality when performing part editing. In this work, to solve these problems, we first propose a novel four-stage point cloud part editing process. Then based on this process and two new strategies: feature disentanglement and constraint, we propose a part editing model SGAS. It can realize various part editing operations. By introducing metrics from multimodal completion and proposing a new metric TMDS, we establish quantitative evaluations for diversity part editing. In addition, SGAS can be pruned to realize unsupervised part-aware point cloud generation. Experiments show that it performs better than previous methods.

Limitation Since we do not design part offset networks for the generated parts but instead utilize the relatively fixed spatial positions of each generated part to ensure good assembly, SGAS can only achieve part editing for objects with relatively consistent prototypes. For example, SGAS cannot handle part editing for both ceiling lamps and table lamps simultaneously as the spatial order of the parts is opposite. In addition, we also find that the performance of SGAS is limited by the pre-trained autoencoders. The embedded features are better when the parts are normalized. Therefore, it will be beneficial to first generate normalized parts and then design part offset networks to align them in the future.

Acknowledgments

This work was supported by National Natural Science Fund of China (62176064). Cheng Jin is the corresponding author.

References

Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; and Guibas, L. J. 2017. Learning Representations and Generative Models For 3D Point Clouds. *arXiv preprint arXiv:1707.02392*.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, 214–223.

Arora, H.; Mishra, S.; Peng, S.; Li, K.; and Mahdavi-Amiri, A. 2022. Multimodal Shape Completion via Implicit Maximum Likelihood Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2958–2967.

Chen, X.; Chen, B.; and Mitra, N. J. 2020. Unpaired Point Cloud Completion on Real Scans using Adversarial Training. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

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 (ICCV)*.

Cheng, A.-C.; Li, X.; Liu, S.; Sun, M.; and Yang, M.-H. 2022. Autoregressive 3d shape generation via canonical mapping. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, 89–104.

Cheng, Y.-C.; Lee, H.-Y.; Tuyakov, S.; Schwing, A.; and Gui, L. 2023. SDFusion: Multimodal 3D Shape Completion, Reconstruction, and Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.

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 (ICCV)*.

Fan, H.; Su, H.; and Guibas, L. J. 2017. A Point Set Generation Network for 3D Object Reconstruction From a Single Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gal, R.; Bermano, A.; Zhang, H.; and Cohen-Or, D. 2021. MRGAN: Multi-Rooted 3D Shape Representation Learning With Unsupervised Part Disentanglement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2039–2048.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, volume 30.

Huang, X.; and Belongie, S. 2017. Arbitrary Style Transfer in Real-Time With Adaptive Instance Normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Hui, K.-H.; Li, R.; Hu, J.; and Fu, C.-W. 2022. Neural Template: Topology-Aware Reconstruction and Disentangled Generation of 3D Meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18572–18582.

Jiang, W.; and Daniilidis, K. 2022. Probabilistic Shape Completion by Estimating Canonical Factors with Hierarchical VAE. *arXiv preprint arXiv:2212.03370*.

Jones, R. K.; Barton, T.; Xu, X.; Wang, K.; Jiang, E.; Guerrero, P.; Mitra, N. J.; and Ritchie, D. 2020. ShapeAssembly: Learning to Generate Programs for 3D Shape Structure Synthesis. *ACM Transactions on Graphics (TOG), Siggraph Asia 2020*, 39(6): Article 234.

Li, J.; Niu, C.; and Xu, K. 2020. Learning Part Generation and Assembly for Structure-Aware Shape Synthesis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07): 11362–11369.

Li, J.; Xu, K.; Chaudhuri, S.; Yumer, E.; Zhang, H.; and Guibas, L. 2017. GRASS: Generative Recursive Autoencoders for Shape Structures. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, 36(4).

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 (Proc. SIGGRAPH)*, 40(4).

Li, S.; Liu, M.; and Walder, C. 2022. EditVAE: Unsupervised Parts-Aware Controllable 3D Point Cloud Shape Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2): 1386–1394.

Li, Y.; and Baciu, G. 2022. SG-GAN: Adversarial Self-Attention GCN for Point Cloud Topological Parts Generation. *IEEE Transactions on Visualization and Computer Graphics*, 28(10): 3499–3512.

Liu, J.; Snodgrass, S.; Khalifa, A.; Risi, S.; Yannakakis, G. N.; and Togelius, J. 2021. Deep learning for procedural content generation. *Neural Computing and Applications*, 33(1): 19–37.

Mittal, P.; Cheng, Y.-C.; Singh, M.; and Tulsiani, S. 2022. AutoSDF: Shape Priors for 3D Completion, Reconstruction and Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 306–315.

Mo, K.; Guerrero, P.; Yi, L.; Su, H.; Wonka, P.; Mitra, N.; and Guibas, L. 2019a. StructureNet: Hierarchical Graph Networks for 3D Shape Generation. *ACM Transactions on Graphics (TOG), Siggraph Asia 2019*, 38(6): Article 242.

Mo, K.; Guerrero, P.; Yi, L.; Su, H.; Wonka, P.; Mitra, N. J.; and Guibas, L. J. 2020. StructEdit: Learning Structural Shape Variations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mo, K.; Zhu, S.; Chang, A. X.; Yi, L.; Tripathi, S.; Guibas, L. J.; and Su, H. 2019b. PartNet: A Large-Scale Benchmark

for Fine-Grained and Hierarchical Part-Level 3D Object Understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Öngün, C.; and Temizel, A. 2020. LPMNet: Latent part modification and generation for 3D point clouds. *Comput. Graph.*, 96: 1–13.

Postels, J.; Liu, M.; Spezialetti, R.; Gool, L. V.; and Tombari, F. 2021. Go with the Flows: Mixtures of Normalizing Flows for Point Cloud Generation and Reconstruction. In *2021 International Conference on 3D Vision (3DV)*, 1249–1258.

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 (CVPR)*.

Schor, N.; Katzir, O.; Zhang, H.; and Cohen-Or, D. 2019. CompoNet: Learning to Generate the Unseen by Part Synthesis and Composition. In *The IEEE International Conference on Computer Vision (ICCV)*.

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 (ICCV)*.

van den Oord, A.; Vinyals, O.; and kavukcuoglu, k. 2017. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems*, volume 30.

Wang, H.; Schor, N.; Hu, R.; Huang, H.; Cohen-Or, D.; and Huang, H. 2018. Global-to-Local Generative Model for 3D Shapes. *ACM Trans. Graph.*, 37(6).

Wang, K.; Guerrero, P.; Kim, V. G.; Chaudhuri, S.; Sung, M.; and Ritchie, D. 2022. The shape part slot machine: Contact-based reasoning for generating 3D shapes from parts. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, 610–626.

Wu, C.; Zheng, J.; Pfrommer, J.; and Beyerer, J. 2023. Attention-based Part Assembly for 3D Volumetric Shape Modeling. *arXiv preprint arXiv:2304.10986*.

Wu, R.; Chen, X.; Zhuang, Y.; and Chen, B. 2020a. Multimodal shape completion via conditional generative adversarial networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, 281–296.

Wu, R.; Zhuang, Y.; Xu, K.; Zhang, H.; and Chen, B. 2020b. PQ-NET: A Generative Part Seq2Seq Network for 3D Shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wu, Z.; Wang, X.; Lin, D.; Lischinski, D.; Cohen-Or, D.; and Huang, H. 2019. SAGNet: Structure-Aware Generative Network for 3D-Shape Modeling. *ACM Trans. Graph.*, 38(4).

Yan, X.; Lin, L.; Mitra, N. J.; Lischinski, D.; Cohen-Or, D.; and Huang, H. 2022. ShapeFormer: Transformer-Based Shape Completion via Sparse Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6239–6249.

Yang, Y.; Feng, C.; Shen, Y.; and Tian, D. 2018. FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I.-C.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; and Guibas, L. 2016. A Scalable Active Framework for Region Annotation in 3D Shape Collections. *ACM Trans. Graph.*, 35(6).

Yin, K.; Chen, Z.; Chaudhuri, S.; Fisher, M.; Kim, V. G.; and Zhang, H. 2020. Coalesce: Component assembly by learning to synthesize connections. In *2020 International Conference on 3D Vision (3DV)*, 61–70.

Zhang, D.; Choi, C.; Kim, J.; and Kim, Y. M. 2021. Learning to Generate 3D Shapes with Generative Cellular Automata. In *International Conference on Learning Representations*.

Zhang, K.; Yang, X.; Wu, Y.; and Jin, C. 2022. Attention-Based Transformation from Latent Features to Point Clouds. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3): 3291–3299.

Zhao, Y.; Zhou, Y.; Chen, R.; Hu, B.; and Ai, X. 2021. MM-Flow: Multi-Modal Flow Network for Point Cloud Completion. In *Proceedings of the 29th ACM International Conference on Multimedia*, 3266–3274.

Zhou, L.; Du, Y.; and Wu, J. 2021. 3D Shape Generation and Completion Through Point-Voxel Diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 5826–5835.

Zhu, C.; Xu, K.; Chaudhuri, S.; Yi, L.; Guibas, L. J.; and Zhang, H. 2020. AdaCoSeg: Adaptive Shape Co-Segmentation With Group Consistency Loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhuang, Y. 2022. Progressive Multimodal Shape Generation via Contextual Part Reasoning. In *2022 The 6th International Conference on Machine Learning and Soft Computing*, 173–178.

Zou, C.; Yumer, E.; Yang, J.; Ceylan, D.; and Hoiem, D. 2017. 3D-PRNN: Generating Shape Primitives With Recurrent Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.