

# SUPPLEMENTARY MATERIALS FOR SEALION: SEMANTIC PART-AWARE LATENT POINT DIFFUSION MODELS FOR 3D GENERATION

**Anonymous authors**

Paper under double-blind review

## 1 PSEUDO CODE OF USING SEALION AS EDITING TOOL

As discussed in Section 3.3 of the main paper, SeaLion can serve as a tool for part-aware 3D shape editing. The related pseudo code is provided below:

---

**Algorithm 1** Part-aware 3D shape editing using SeaLion.

---

```

1: Input: Point cloud  $x$  consisting of  $n$  points, segmentation labels  $y$ , desired fix-shape part  $p$ .
2: Output: Novel generated point cloud  $x_0$  with preserved fix-shape part  $p$  and variation in the
   remaining parts, along with the updated segmentation labels  $y_0$ .
3:  $\text{mask}_p \leftarrow (y == p)$   $\triangleright$  Define a boolean mask to select points belonging to part  $p$ 
4:  $z_0 \leftarrow \phi_z(x)$ 
5:  $h_0 \leftarrow \phi_h(x, y, z_0)$ 
6:  $y_\tau \leftarrow y$   $\triangleright \tau < T$ 
7: Perturb  $h_0$  for  $\tau$  steps to  $h_\tau$ 
8: for  $t \leftarrow \tau$  to 1 do
9:    $h_{t-1}, y_{t-1} \leftarrow \epsilon_h(h_t, t, z_0)$ 
10:   $y_{t-1} \leftarrow \alpha \cdot y_{t-1} + (1 - \alpha) \cdot y_t$   $\triangleright$  EMA smooth
11:   $\text{mask}_p^{t-1} \leftarrow ((1 - \text{mask}_p) \odot y_{t-1}) == p$ 
12:   $n_p^{t-1} \leftarrow \sum \text{mask}_p^{t-1}$ 
13:  if  $n_p^{t-1} > 0$  then  $\triangleright$  Substitute the latent points in the remaining part but predicted as
   fix-shape part  $p$ 
14:     $\text{mask}_{\text{others}}^{t-1} \leftarrow ((1 - \text{mask}_p) \odot y_{t-1}) \neq p$ 
15:    Extract non-zero indices in  $\text{mask}_{\text{others}}^{t-1}$ , randomly sample  $n_p^{t-1}$  elements and then create a
   boolean mask for substitution  $\text{mask}_{\text{resample}}^{t-1}$ 
16:     $h_{t-1}[\text{mask}_p^{t-1}] \leftarrow h_{t-1}[\text{mask}_{\text{resample}}^{t-1}]$ 
17:     $y_{t-1}[\text{mask}_p^{t-1}] \leftarrow y_{t-1}[\text{mask}_{\text{resample}}^{t-1}]$ 
18:  end if
19:  Perturb  $h_0$  for  $t$  steps to  $h_t^*$ 
20:   $h_{t-1} \leftarrow \text{mask}_p \odot h_{t-1}^* + (1 - \text{mask}_p) \odot h_{t-1}$ 
21:   $y_{t-1} \leftarrow \text{mask}_p \odot y + (1 - \text{mask}_p) \odot y_{t-1}$ 
22: end for
23:  $x_0 \leftarrow \xi_h(h_0, y_0, z_0)$ 
24: Return  $x_0, y_0$ 

```

---

## 2 EVALUATION METRICS

Given a generated dataset  $\mathcal{G} = \{x^g | x^g \in \mathbb{R}^{n \times 3}\}$  and a real dataset  $\mathcal{R} = \{x^r | x^r \in \mathbb{R}^{n \times 3}\}$ , both consist of point clouds with  $n$  points.  $D(\cdot)$  is either the Chamfer distance or earth mover’s distance to measure the distance between two point clouds.

**Coverage (COV)**

$$\text{COV}(\mathcal{G}, \mathcal{R}) = \frac{|\{\arg \min_{x_r \in \mathcal{R}} D(x_g, x_r) | x_g \in \mathcal{G}\}|}{|\mathcal{R}|} \quad (1)$$

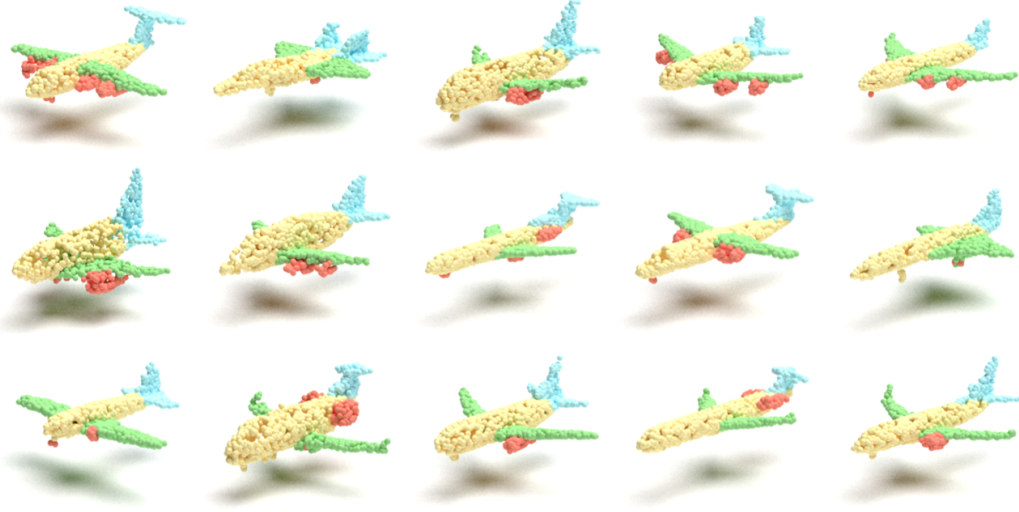


Figure 1: Generated point clouds of airplane class.

**Minimum matching distance (MMD)**

$$\text{MMD}(\mathcal{G}, \mathcal{R}) = \frac{1}{|\mathcal{R}|} \sum_{x_r \in \mathcal{R}} \min_{x_g \in \mathcal{G}} D(x_g, x_r) \quad (2)$$

**1-nearest neighbor accuracy (1-NNA)**

$$\text{1-NNA}(\mathcal{G}, \mathcal{R}) = \frac{\sum_{x_g \in \mathcal{G}} \mathbb{1}(N_{x_g} \in \mathcal{G}) + \sum_{x_r \in \mathcal{R}} \mathbb{1}(N_{x_r} \in \mathcal{R})}{|\mathcal{G}| + |\mathcal{R}|}, \quad (3)$$

where  $\mathbb{1}[\cdot]$  is the indicator function,  $N_{x_g}$  is the nearest neighbor of  $x_g$  in the set  $\mathcal{R} \cup \mathcal{G} \setminus \{x_g\}$ , with the same applying to  $N_{x_r}$ .

**Inter-part score (snapping metric, SNAP)**

The inter-part score, the snapping metric (SNAP) (Nakayama et al., 2023), measures the connection tightness between two contacting parts in a object by computing the Chamfer distance between their closet  $N_{\text{SNAP}}$  points, e.g.  $N_{\text{SNAP}} = 30$ . For the point cloud  $x$ , the SNAP score is calculated as follows:

$$\text{SNAP}(x) = \frac{1}{|P|} \sum_{p_1 \in P} \min_{x_{p_2} \in \mathcal{X}_{p_1}} \text{Chamfer}\{N_{x_{p_2}}^{(N_{\text{SNAP}})}(x_{p_1}), N_{x_{p_1}}^{(N_{\text{SNAP}})}(x_{p_2})\}, \quad (4)$$

where  $\mathcal{X}_{p_1}$  denotes the connected parts to  $x_{p_1}$ , e.g. if  $x_{p_1}$  is the car body,  $\mathcal{X}_{p_1}$  represents the contacting parts to the car body, {roof, hood, wheel}.  $N_{x_{p_2}}^{(N_{\text{SNAP}})}(x_{p_1})$  refers to the  $N_{\text{SNAP}}$  nearest points in part  $x_{p_2}$  to part  $x_{p_1}$ .

**3 EXPERIMENTAL RESULTS**

Some of the generated point clouds of airplane, car, chair, guitar, lamp, and table categories are demonstrated in Figure 1, 2, 3, 4, 5, and 6, respectively.

**4 MODEL ARCHITECTURE DETAILS**

Details about the hyper-parameters of global encoder  $\phi_z$ , global diffusion module  $\epsilon_z$ , point-level encoder  $\phi_h$ , point-level decoder  $\xi_h$ , and point-level diffusion module  $\epsilon_h$  are listed in Table 1, 2, 3, 4, and 5, respectively. PVConv, SA, GA, and FP refer to point-voxel convolutions modules (Liu et al., 2019), set abstraction layers (Qi et al., 2017), global attention layers, and feature propagation layers (Qi et al., 2017), respectively.



Figure 2: Generated point clouds of car class.

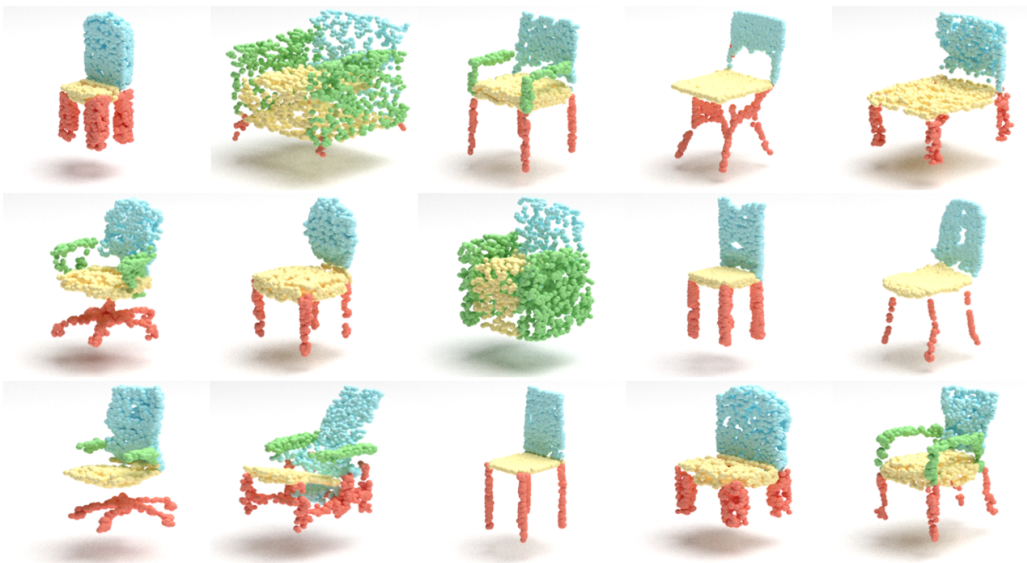


Figure 3: Generated point clouds of chair class.

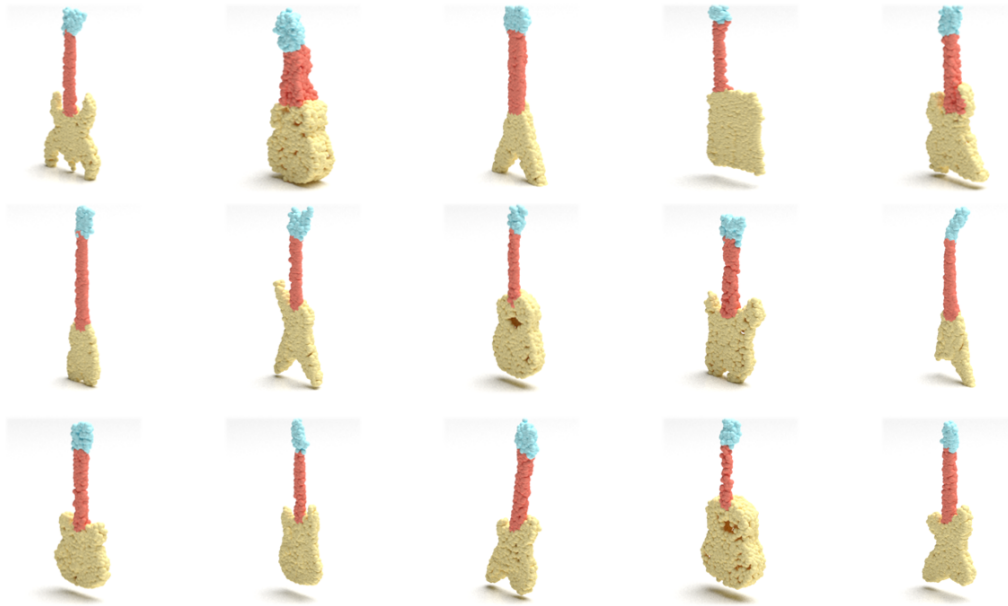


Figure 4: Generated point clouds of guitar class.



Figure 5: Generated point clouds of lamp class.

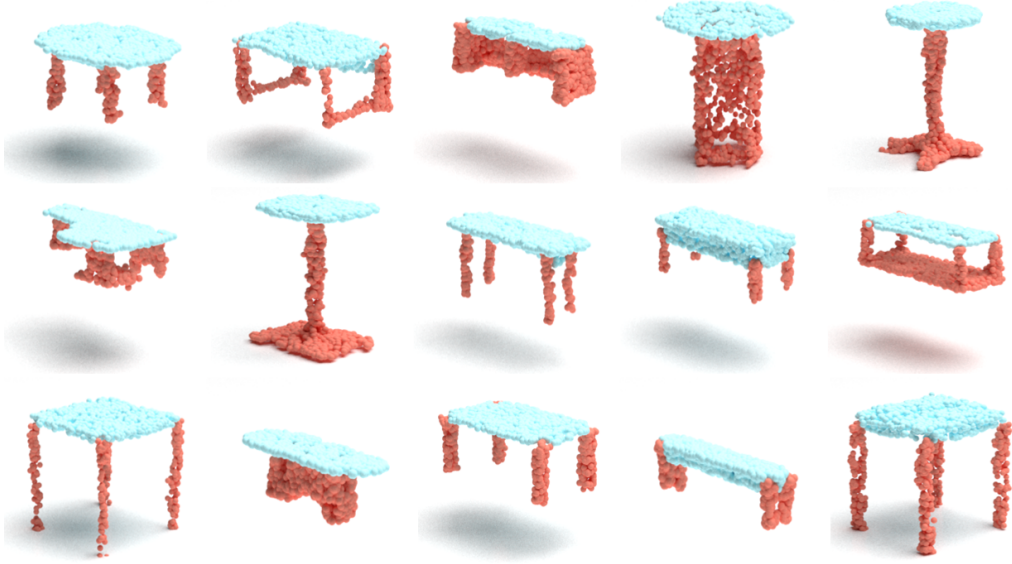


Figure 6: Generated point clouds of table class.

Input	point clouds ( $2048 \times 3$ )		
Output	global latent ( $1 \times 128$ )		
		Layer 1	Layer 2
PVConv	layers	2	1
	hidden dimensions	32	32
	voxel grid size	32	16
SA	grouper center	1024	256
	grouper radius	0.1	0.2
	grouper neighbors	32	32
	MLP layers	2	2
	MLP output dimensions	32, 32	32, 64
Output layer	MLP layers	2	
	MLP output dimensions	128, 128	

Table 1: Hyper-parameters of the global encoder  $\phi_z$ .

Input	global latent ( $1 \times 128$ ), diffusion time step $t$	
Output	predicted noise on global latent ( $1 \times 128$ )	
Input linear layer	output dimension	2048
Time embedding layer	sinusoidal embedding dimension	128
	MLP layers	2
	MLP output dimensions	512, 2048
Stacked ResNet	MLP layers	2
	MLP output dimensions	2048, 2048
	SE MLP layers	2
	SE MLP output dimensions	256, 2048
Output linear layer	output dimension	128

Table 2: Hyper-parameters of the global diffusion  $\epsilon_z$ .

Input	point clouds ( $2048 \times 3$ ), segmentation labels ( $2048 \times c$ ), global latent ( $1 \times 128$ )				
Output	point-level latent ( $2048 \times 4$ )				
		Layer 1	Layer 2	Layer 3	Layer 4
PVConv	layers	2	1	1	-
	hidden dimensions	32	64	128	-
	voxel grid size	32	16	8	-
SA	grouper center	1024	256	64	16
	grouper radius	0.1	0.2	0.4	0.8
	grouper neighbors	32	32	32	32
	MLP layers	2	2	2	3
	MLP output dimensions	32, 32	64, 128	128, 256	128, 128, 128
GA	hidden dimensions	32	128	256	128
	attention heads	8	8	8	8
FP	MLP layers	3	2	2	2
	MLP output dimensions	128, 128, 64	128, 128	128, 128	128, 128
PVConv	layers	2	2	3	3
	hidden dimensions	64	128	128	128
	voxel grid size	32	16	8	8

Table 3: Hyper-parameters of the point-level encoder  $\phi_h$ . Note: layer 1 refers to the shallowest layer and layer 4 refers to the deepest layer,  $c$  denotes the number of parts.

Input	point-level latent ( $2048 \times 4$ ), segmentation labels ( $2048 \times c$ ), global latent ( $1 \times 128$ )				
Output	point cloud ( $2048 \times 3$ )				
		Layer 1	Layer 2	Layer 3	Layer 4
PVConv	layers	2	1	1	-
	hidden dimensions	32	64	128	-
	voxel grid size	32	16	8	-
SA	grouper center	1024	256	64	16
	grouper radius	0.1	0.2	0.4	0.8
	grouper neighbors	32	32	32	32
	MLP layers	2	2	2	3
	MLP output dimensions	32, 64	64, 128	128, 256	128, 128, 128
GA	hidden dimensions	64+c	128+c	256+c	128+c
	attention heads	8	8	8	8
FP	MLP layers	3	2	2	2
	MLP output dimensions	128, 128, 64	128, 128	128, 128	128, 128
PVConv	layers	2	2	3	3
	hidden dimensions	64	128	128	128
	voxel grid size	32	16	8	8
Output layer	MLP layers	2			
	MLP output dimensions	128, 3			

Table 4: Hyper-parameters of the point-level decoder  $\xi_h$ . Note: layer 1 refers to the shallowest layer and layer 4 refers to the deepest layer,  $c$  denotes the number of parts.

Input	point-level latent ( $2048 \times 4$ ), diffusion time step $t$ , global latent ( $1 \times 128$ )				
Output	predicted noise on point-level latent ( $2048 \times 4$ ), predicted segmentation labels ( $2048 \times c$ )				
Time embedding	sinusoidal dimensions	64			
	MLP layers	2			
	MLP output dimensions	64, 64			
		Layer 1	Layer 2	Layer 3	Layer 4
PVConv	layers	2	1	1	-
	hidden dimensions	32	64	128	-
	voxel grid size	32	16	8	-
SA	grouper center	1024	256	64	16
	grouper radius	0.1	0.2	0.4	0.8
	grouper neighbors	32	32	32	32
	MLP layers	2	2	2	3
	MLP output dimensions	32, 64	64, 128	128, 256	128, 128, 128
GA	hidden dimensions	64	128	256	128
	attention heads	8	8	8	8
FP (noise)	MLP layers	3	2	2	2
	MLP output dimensions	128, 128, 64	128, 128	128, 128	128, 128
PVConv (noise)	layers	2	2	3	3
	hidden dimensions	64	128	128	128
	voxel grid size	32	16	8	8
Output layer (noise)	MLP layers	2			
	MLP output dimensions	128, 4			
FP (segmentation)	MLP layers	3	2	2	2
	MLP output dimensions	128, 128, 64	128, 128	128, 128	128, 128
PVConv (segmentation)	layers	2	2	3	3
	hidden dimensions	64	128	128	128
	voxel grid size	32	16	8	8
Output layer (segmentation)	MLP layers	2			
	MLP output dimensions	128, c			

Table 5: Hyper-parameters of the point-level diffusion  $\epsilon_h$ . Note: layer 1 refers to the shallowest layer and layer 4 refers to the deepest layer,  $c$  denotes the number of parts.

## REFERENCES

- Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.
- George Kiyohiro Nakayama, Mikaela Angelina Uy, Jiahui Huang, Shi-Min Hu, Ke Li, and Leonidas Guibas. Diffacto: Controllable part-based 3d point cloud generation with cross diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14257–14267, 2023.
- Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.