Spatially Invariant Unsupervised 3D Object-Centric Learning and Scene Decomposition

Tianyu Wang[1 [0000−0001−9032−8488]], Miaomiao Liu[1 [0000−0001−6485−3510]], and Kee Siong Ng[1 [0000−0003−0701−8783]]

Australian National University, Canberra ACT 2601, AU
{Tianyu.Wang2,miaomiao.liu,KeeSiong.ng}@anu.edu.au

Abstract. We tackle the problem of object-centric learning on point clouds, which is crucial for high-level relational reasoning and scalable machine intelligence. In particular, we introduce a framework, **SPAIR3D**, to factorize a 3D point cloud into a spatial mixture model where each component corresponds to one object. To model the spatial mixture model on point clouds, we derive the *Chamfer Mixture Loss*, which fits naturally into our variational training pipeline. Moreover, we adopt an object-specification scheme that describes each object's location relative to its local voxel grid cell. Such a scheme allows **SPAIR3D** to model scenes with an arbitrary number of objects. We evaluate our method on the task of unsupervised scene decomposition. Experimental results demonstrate that **SPAIR3D** has strong scalability and is capable of detecting and segmenting an unknown number of objects from a point cloud in an unsupervised manner.

Keywords: Deep Generative Model, Variational Inference, Unsupervised Scene Understanding

1 Introduction

3D scenes can exhibit complex and combinatorially large observation spaces even when there are only a few basic elements. Motivated in part by cognitive psychology studies [26] that suggest human brains organize observations at an object level, recent advances in physical prediction [8], and the superior robustness demonstrated by object-oriented reinforcement learning agent[12,27], we tackle in this paper the problem of deep object-centric learning on point clouds, which is crucial for high-level relational reasoning and scalable machine intelligence.

There is a good body of existing literature on unsupervised object-centric generative models for images and videos. The spatial mixture models are widely adopted to model observations in an object-oriented way [5,10,14,16,30]. These approaches effectively define objectness as a region with strong appearance correlations, and Variational Autoencoders (VAE) [19,28] play a critical role in exploiting such correlations. More precisely, the encoder-decoder structure of

VAE effectively creates an information bottleneck [2,6,36] limiting the amount of information passing through. To reconstruct the observation under a limited information budget, highly correlated information must be encoded together. Thus, objectness emerges from the encoding strategy. The above-mentioned papers mainly exploit appearance correlations on objects that are colored uniformly. In this paper, we show that this paradigm is also applicable to structural correlations conveyed by point clouds without appearance information, as long as we can overcome some irregularities in point cloud data as described in section 3. Specifically, inspired by SPAIR [10], we propose in this paper a VAE-based model named **Sp**atially Invariant **A**ttend, **I**nfer, **R**epeat in **3D** (SPAIR3D), a model that generates spatial mixture distributions on point clouds to discover 3D objects in static scenes. Here we summarize the key contributions:

- We propose, to the best of our knowledge, the first unsupervised object-centric learning pipeline for point cloud data, named SPAIR3D.
- We also propose a new *Chamfer Mixture Loss* function tailored for learning mixture models over point cloud data with a novel graph neural network that can be used to model and generate a variable number of 3D points.
- We provide qualitative and quantitative results to show that SPAIR3D learns meaningful object-centric representation and decomposes point clouds scene with an arbitrary number of objects in an object-oriented manner.

2 Related Work

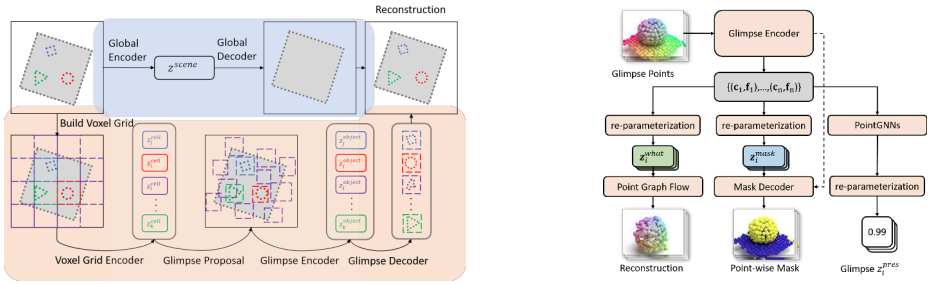
Generative Unsupervised Object-centric Learning. Unsupervised object-centric learning based on generative models has attracted increasing attention in recent times. Such approaches focus on joint object representation-learning and scene decomposition based on single or multiple views [5,9,10,13,16,29,30,35]. A spatial Gaussian mixture model is typically defined on 2D images consisting of K mixture components that correspond to K objects. Each component spans the entire image and places an isotropic Gaussian on the RGB value of all pixels with a predicted mean and a constant covariance. Each component also assigns each pixel a non-negative mixing weight that sums to one across all components. The definition can be easily extended to voxel and neural radiance fields.

Under the spatial mixture model formulation, different inference methods are proposed. IODINE [16] employs iterative amortized inference to refine latent variable posteriors for all components in parallel. GENESIS [13] and MONET [5] sequentially infer the latent representation, one component at a time. Slot attention [31] and Neural Expectation Maximization (NEM) [17] can be regarded as differentiable clustering algorithms.

Instead of treating each component of the mixture model as a full-scale observation, **A**ttend, **I**nfer, **R**epeat (AIR) [14] confines the extent of each component to a local region. In AIR, one network is trained to propose a set of candidate object regions in the form of 2D bounding boxes. Each region is then cropped out and processed by a VAE. The final reconstruction is obtained by placing

the reconstructed patches back in the inferred locations. Pixels that are not covered by any patches are deemed background. While AIR fails in scenes of dense objects, SPAIR [10] addresses the challenge with a grid spatial attention mechanism with which bounding boxes are proposed locally from each grid cell. This extension is also proven effective in object-tracking tasks [11]. By confining the extent of each component, constraints on maximum object sizes are imposed. SPACE [30] employs MONET to model background components that are normally much larger than foreground objects.

Graph Neural Network for Point Cloud Generation. Generative models such as VAEs [15] and generative adversarial networks [1] have been successfully used for point cloud generation but with a pre-defined number of points per-object. It is shown that a point cloud generation process can be modeled as a latent variable conditioned Markov chain [32]. PointFlow [38] is a normalizing flow-based approach that models object shapes as continuous distributions and allows the generation of a variable number of points. It could not be naturally integrated into our framework due to the need for an ODE solver.



(a) Structure of SPAIR3D (b) Structure of Glimpse VAE

Fig. 1: (a) Structure of SPAIR3D. For clarity, we adopt 2D abstraction and use colors to highlight important correspondence. (b) Structure of Glimpse VAE. Glimpse encoder encodes foreground glimpses and produce $\mathbf{z}_i^{what}$, $\mathbf{z}_i^{mask}$ and z_i^{pres} for each glimpse. Point Graph Decoder takes $\mathbf{z}_i^{what}$ and reconstructs input points (left branch). Mask Decoder takes $\mathbf{z}_i^{mask}$ and generates masks for each point (middle branch). The dashed line represents the dependency on the coordinates of the intermediate points in the hierarchy and $\mathcal{G}_i$. Multi-layer PointGNN enable message passing between $(\mathbf{c}_i, \mathbf{f}_i)$ and produces z_i^{pres} (right branch).

3 SPAIR3D

While the application of generative-model-based object-centric learning on image [5,16], voxel [18], and mesh [18] shows encouraging results, its application on point cloud has not been explored up till now. Unlike point cloud data, the reconstructions of images, and voxels are all coordinate-dependent. For each

mixture component, given a coordinate, a mixing weight (defining mask) and a feature vector (RGB value) are generated at that coordinate to form a well-defined mixture model. For image data, the coordinate dependency can be implicitly embedded in the network structure since the input and output are of fixed sizes [21]. The coordinate thus provides the correspondence between input and reconstruction, inducing a natural likelihood function.

However, a point cloud takes the form of an unordered set with irregular structures. Each point cloud may have a varying number of points. More importantly, the point coordinates carrying structural information becomes the reconstruction target, and there is usually no natural correspondence between the input and the reconstruction. While *Chamfer Distance* commonly serves as a loss function for point cloud reconstruction, it does not support mixture model formulation directly. Such data irregularity makes defining a mixture model over point cloud a non-trivial task.

To overcome the issues outlined above, we extend the SPAIR framework and introduce **SPAIR3D**, a deep generative model for 3D object-centric learning and 3D scene decomposition via object-centric point-cloud generation. There are two main reasons for choosing the SPAIR framework over others. Firstly, as a consequence of the lack of correspondence between input and reconstruction, the likelihood computation commonly involves a bi-directional matching between the generated point cloud and the ground truth, leading to quadratic time complexity. The local object proposal and reconstruction mechanism allow us to confine the matching computation in each local region, which significantly reduces the algorithm's time complexity (see Supp. Sec. 1 for further analysis).

Secondly, 3D point cloud reconstruction commonly requires the target object to be centered [32,38]. While it is straightforward to center objects in single-object reconstruction tasks, it is difficult to center all objects in the same coordinate system in the 3D scene reconstruction setup. In contrast, thanks to the local object proposal mechanism proposed in this paper, each object can be naturally centered in a local coordinate system.

Below we first introduce our generative model formulation over key latent variables (§3.1). Then we detail the inference model implementation (§3.3). We discuss the particular challenges arising in handling a varying number of points with a novel *Chamfer Mixture Loss* (§3.2) and *Point Graph Decoder* (§3.3).

3.1 Local Object Proposal and Generative Model

As shown in Fig. 1a, SPAIR3D first divides a 3D scene into a spatial attention voxel grid with possible empty voxel cells covering no points. We discard empty cells and associate a bounding box with each non-empty voxel cell. The set of input points captured by a bounding box is termed an *object glimpse*. Besides object glimpses, SPAIR3D also defines a *scene glimpse* covering all points in an input scene. Later, we show that we encode and reconstruct each glimpse and generate a mixing weight on each point to form a probability mixture model.

Similar to SPAIR, each grid cell generates posterior distributions over a set of latent variables defined as $\mathbf{z}_i^{cell} = \{\mathbf{z}_i^{where}, \mathbf{z}_i^{apothem}\}$, where $\mathbf{z}_i^{where} \in \mathbb{R}^3$ encodes

the relative position of the center of the i^{th} bounding box to the center of the i^{th} cell, $\mathbf{z}_i^{apothem} \in \mathbb{R}^3$ encodes the apothem of the bounding box. Thus, each $\mathbf{z}_i^{cell}$ induces one object glimpse associated with the i^{th} cell. Each object glimpse is then associated with posterior distributions over latent variables specified as $\mathbf{z}_i^{object} = \{\mathbf{z}_i^{what}, \mathbf{z}_i^{mask}, z_i^{pres}\}$, where $\mathbf{z}_i^{what} \in \mathbb{R}^A$ encodes the structure information of the corresponding object glimpse, $\mathbf{z}_i^{mask} \in \mathbb{R}^B$ encodes the mask for each point in the glimpse, $z_i^{pres} \in \{0, 1\}$ is a binary variable indicating whether the proposed object should exist ($z_i^{pres} = 1$) or not ($z_i^{pres} = 0$). The scene glimpse is associated with only one latent variable $\mathbf{z}^{scene} = \{\mathbf{z}_0^{what}\}$. We assume z_i^{pres} follows a Bernoulli distribution. The posteriors and priors of other latent variables are all set to isotropic Gaussian distributions.

Given latent representations of objects and the scene, the complete likelihood for a point cloud $\mathcal{X}$ is $p(\mathcal{X}) = \int_{\mathbf{z}} p(\mathbf{z})p(\mathcal{X}|\mathbf{z})d\mathbf{z}$, where $\mathbf{z} = (\bigcup_i \mathbf{z}_i^{cell}) \cup (\bigcup_i \mathbf{z}_i^{object}) \cup \mathbf{z}^{scene}$. As maximizing the objective $p(\mathcal{X})$ is intractable, we resort to the variational inference to maximize its evidence lower bound (ELBO).

3.2 Chamfer Mixture Loss

Unlike generative model-based unsupervised 2D segmentation methods that reconstruct the pixel-wise appearance conditioning on its spatial coordinate, the reconstruction of a point cloud lost its point-wise correspondence to the original point cloud. *Chamfer distance* is commonly adopted to measure the discrepancy between the generated point cloud ($\hat{\mathcal{X}}$) and the input point cloud ($\mathcal{X}$). Formally, *Chamfer distance* is defined by $d_{CD}(\mathcal{X}, \hat{\mathcal{X}}) = \sum_{x \in \mathcal{X}} \min_{\hat{x} \in \hat{\mathcal{X}}} \|x - \hat{x}\|_2^2 + \sum_{\hat{x} \in \hat{\mathcal{X}}} \min_{x \in \mathcal{X}} \|x - \hat{x}\|_2^2$. We refer to the first and the second term as the forward loss and the backward loss, respectively.

Unfortunately, the *Chamfer distance* does not fit naturally into the mixture model framework. To get around that, we propose a *Chamfer Mixture Loss* (CML) tailored for training probability mixture models defined on point clouds. The *Chamfer Mixture Loss* is composed of a *forward likelihood* and a *backward regularization* corresponding to the forward and backward loss, respectively.

Denote the i^{th} glimpse as $\mathcal{G}_i$, $i \in \{0, \ldots, n\}$ and its reconstruction as $\hat{\mathcal{G}}_i$, $i \in \{0, \ldots, n\}$. Specifically, we treat the scene glimpse as the 0^{th} glimpse that contains all input points, that is, $\mathcal{G}_0 = \mathcal{X}$. Note that one input point can be a member of multiple glimpses. Below we use $\mathcal{N}(x|\mu, \sigma)$ to denote the probability density value of point x evaluated at a Gaussian distribution of mean μ and variance σ. For each input point x in the i^{th} glimpse, the glimpse-wise forward likelihood of that point is defined as $\mathcal{L}_i^F(x) = \frac{1}{u_i} \max_{\hat{x} \in \hat{\mathcal{G}}_i} \mathcal{N}(x|\hat{x}, \sigma_c)$, where $u_i = \int_{x \in \mathcal{X}} \max_{\hat{x} \in \hat{\mathcal{G}}_i} \mathcal{N}(x|\hat{x}, \sigma_c)\mathrm{d}x$ is the normalizer and σ_c is a hyperparameter. For each glimpse $\mathcal{G}_i$, $i \in \{0, \ldots, n\}$, $\alpha_i^x \in [0, 1]$ defines a mixing weight for point x in the glimpse and $\sum_{i=0}^n \alpha_i^x = 1$. In particular, α_i^x, $i \in \{1, \ldots, n\}$, is determined by $\alpha_i^x = \frac{z_i^{pres} \pi_i^x}{\sum_{j=1}^n z_j^{pres} \pi_j^x} z_i^{pres} \pi_i^x$, where π_i^x is the predicted mask value and $\pi_i^x = 0$ if $x \notin \mathcal{G}_i$. The mixing weight for the scene layout points completes the distribution through $\alpha_0^x = 1 - \sum_{i=1}^n \alpha_i^x$ for $x \in \mathcal{G}_0$. Thus, the final mixture model for an

input point x is $\mathcal{L}^F(x) = \sum_{i=0}^{n} \alpha_i^x \mathcal{L}_i^F(x)$. The total forward likelihood of $\mathcal{X}$ is then defined as $\mathcal{L}^F(\mathcal{X}) = \prod_{x \in \mathcal{X}} \mathcal{L}^F(x)$.

The forward likelihood alone leads to a trivial sub-optimal solution with $\hat{\mathcal{X}}$ distributed densely and uniformly in the space. To enforce a high-quality reconstruction, we define a backward regularization term. For each predicted point $\hat{x}$, the point-wise backward regularization is $\mathcal{L}^B(\hat{x}) = \max_{x \in \mathcal{G}_{i(\hat{x})}} \mathcal{N}(\hat{x}|x, \sigma_c)$, where $i(\hat{x})$ returns the glimpse index of $\hat{x}$. We denote $x(\hat{x}) = \arg\max_{x \in \mathcal{G}_{i(\hat{x})}} \mathcal{N}(\hat{x}|x, \sigma_c)$ and $\hat{\mathcal{X}} = \bigcup_{i=0}^{n} \hat{\mathcal{G}}_i$. The backward regularization is then defined as $\mathcal{L}^B(\hat{\mathcal{X}}) = \prod_{i=0}^{n} \prod_{\hat{x} \in \hat{\mathcal{G}}_i} \mathcal{L}^B(\hat{x})^{\alpha_i^{x(\hat{x})}}$. The exponential weighting, i.e. $\alpha_i^{x(\hat{x})} \in [0, 1]$, is crucial. As each predicted point $\hat{x} \in \hat{\mathcal{X}}$ belongs to one and only one glimpse, it is difficult to impose a mixture model interpretation on the backward regularization. The exponential weighting encourages the generated points in object glimpse to be close to input points with a high probability belonging to $\mathcal{G}_i$. Combining the forward likelihood and the backward regularization together, we define *Chamfer Mixture Loss* as $\mathcal{L}_{\mathcal{CD}}(\mathcal{X}, \hat{\mathcal{X}}) = \mathcal{L}^F(\mathcal{X}) \cdot \mathcal{L}^B(\hat{\mathcal{X}})$. During inference, the segmentation label for each point x is naturally obtained by $\arg\max_i \alpha_i^x$.

The overall loss function is $\mathcal{L} = -\log \mathcal{L}_{\mathcal{CD}}(\mathcal{X}, \hat{\mathcal{X}}) + \mathcal{L}_{KL}(\mathbf{z}^{cell}, \mathbf{z}^{object}, \mathbf{z}^{scene})$, where $\mathcal{L}_{KL}$ is the KL divergence between the prior and posterior of the latent variables (Supp. Sec. 2 for details). In general, one cannot find a closed-form solution for the normalizer in Chamfer Mixture Loss. However, the experiments below show that we can safely ignore it during optimization.

3.3 Model Structure

We next introduce the encoder and decoder network structure for SPAIR3D. The building blocks are based on graph neural networks and point convolution operations (See Sec. 4 in the Supp. for details).

Encoder network. We design an encoder network $q_\phi(\mathbf{z}|x)$ to obtain the latent representations $\{\mathbf{z}_i^{cell}\}_{i=1}^n$ and $\{\mathbf{z}_i^{object}\}_{i=1}^n$ from a point cloud. To achieve the spatially invariant property, we group one PointConv [37] layer and one PointGNN [34] layer into pairs for message passing and information aggregation among points and between cells.

(a) Voxel Grid Encoding. The voxel-grid encoder takes a point cloud as input and generates for each spatial attention voxel cell $\mathcal{C}_i$ two latent variables $\mathbf{z}_i^{where} \in \mathbb{R}^3$ and $\mathbf{z}_i^{apothem} \in \mathbb{R}^3$ to propose a glimpse $\mathcal{G}_i$ potentially occupied by an object. To better capture the point cloud information in $\mathcal{C}_i$, we build a voxel pyramid within each cell $\mathcal{C}_i$ with the bottom level corresponding to the finest voxel grid. We aggregate information hierarchically using PointConv-PointGNN pairs from bottom to top through each level of the pyramid. For each layer of the pyramid, we aggregate the features of all points and assign them to the point spawned at the center of mass of the voxel cell. Then PointGNN is employed to perform message passing on the radius graph built on all spawned points. The output of the final aggregation block produces $\mathbf{z}_i^{where}$ and $\mathbf{z}_i^{apothem}$ via the re-parametrization trick [28].

We obtain the offset distance of a glimpse center from its corresponding grid cell center using $\Delta g_i = \tanh(\mathbf{z}_i^{where}) \cdot L$, where L is the maximum offset distance. The apothems of the glimpse in the x, y, z direction is given by $\Delta \mathbf{g}_i^{apo} = T(\mathbf{z}_i^{apothem})(\mathbf{r}^{max} - \mathbf{r}^{min}) + \mathbf{r}^{min}$, where $T(\cdot)$ is the sigmoid function and $[\mathbf{r}^{min}, \mathbf{r}^{max}]$ defines the range of apothem.

(b) Glimpse Encoding. The predicted glimpse center offset and the apothems uniquely determine one glimpse for each spatial attention voxel cell. We adopt the same encoder structure to encode each glimpse $\mathcal{G}_i$ into one point $\mathbf{a}_i = (\mathbf{c}_i, \mathbf{f}_i)$, where $\mathbf{c}_i$ is the glimpse center coordinate and $\mathbf{f}_i$ is the glimpse feature vector. We then generate $\mathbf{z}_i^{what}$ and $\mathbf{z}_i^{mask}$ from $\mathbf{a}_i$ via the re-parameterization trick.

The variable z_i^{pres} governs the glimpse rejection process and is crucial to the final decomposition quality. Unlike previous work [10,30], SPAIR3D generates z^{pres} from glimpse features instead of cell features based on our observation that message passing across glimpses provides more benefits in the glimpse-rejection process. To this end, a radius graph is first built on the point set $\{(\mathbf{c}_i, \mathbf{f}_i)\}_{i=1}^n$ to connect nearby glimpse centers, which is followed by multiple PointGNN layers with decreasing output channels to perform local message passing. The z_i^{pres} of each glimpse is then obtained via the re-parameterization trick. Information exchange between nearby glimpses can help avoid over-segmentation that would otherwise occur because of the high dimensionality of point cloud data.

(c) Global Encoding. The global encoding module adopts the same encoder as the object glimpse encoder to encode scene glimpse $\mathcal{G}_0$. The learned latent representation is $\mathbf{z}_0^{what}$ with $z_0^{pres} = 1$.

Decoder network. We now introduce the decoders used for point-cloud and mask generation.

(a) Point Graph Decoder (PGD). Given the $\mathbf{z}_i^{object}$ of each glimpse, the decoder is used for point-cloud reconstruction as well as segmentation-mask generation. In reconstruction, the number of generated points has a direct effect on the magnitudes of the forward and backward terms in the Chamfer Mixture Loss. An unbalanced number of reconstruction points can lead to under- or over-segmentation. To balance the forward likelihood and the backward regularization, the number of predictions for each glimpse should be approximately the same as the number of input points. We propose a graph network based point decoder to allow setting the size of $\hat{\mathcal{X}}$ in run time.

PGD treats the point cloud reconstruction as a point diffusion process [32]. The input to the PGD is a set of 3D points with coordinates sampled from a zero-centered Gaussian distribution, with the population determined by the number of points in the current glimpse. Features of the input points are set uniformly to the latent variable $\mathbf{z}_i^{what}$. PGD is composed of several PointGNN layers, each of which is preceded by a radius graph operation. The output of each PointGNN layer is of dimension $f + 3$, with the first f dimensions interpreted as the updated features and the last 3 dimensions interpreted as the updated 3D coordinates for estimated points. Since we only focus on point coordinates prediction, we set $f = 0$ for the last PointGNN layer.

(b) Mask Decoder. The Mask Decoder decodes $(\mathbf{c}_i, \mathbf{z}_i^{mask})$ to the mask value, $\pi_i^x \in [0, 1]$, of each point within a glimpse $\mathcal{G}_i$. The decoding process follows the exact inverse pyramid structure of the Glimpse Encoder. To be more precise, the mask decoder can access the spatial coordinates of the intermediate aggregation points of the Glimpse Encoder as well as the point coordinates of $\mathcal{G}_i$. During decoding, PointConv is used as deconvolution operation.

Glimpse VAE and Global VAE. The complete Glimpse VAE structure is presented in Fig. 1b. The Glimpse VAE is composed of a Glimpse Encoder, Point Graph Decoder, Mask Decoder and a multi-layer PointGNN network. The Glimpse Encoder takes all glimpses as input and encodes each glimpse $\mathcal{G}_i$ individually and in parallel into feature points $(\mathbf{c}_i, \mathbf{f}_i)$. Via the re-parameterization trick, $\mathbf{z}_i^{what}$ and $\mathbf{z}_i^{mask}$ are then obtained from $\mathbf{f}_i$. From there, we use the Point Graph Decoder to decode $\mathbf{z}_i^{what}$ to reconstruct the input points, and we use the Mask Decoder to decode $\mathbf{z}_i^{mask}$ to assign a mask value for each input point within $\mathcal{G}_i$. Finally, $\mathbf{z}_i^{pres}$ is generated via message passing among neighbour glimpses. All glimpses are processed in parallel. The Global VAE consisting of the Global Encoder and a PGD outputs the reconstructed scene layout.

3.4 Soft Boundary

The prior of $\mathbf{z}^{apothem}$ is set to encourage apothem to shrink so that the size of the glimpses will not be overly large. However, if one point is excluded from one glimpse, its gradient is disconnected from the size and location of the glimpse anymore, and this can lead to over-segmentation. To solve this problem, we introduce a soft boundary weight $b_i^x \in [0, 1]$ which decreases as a point $x \in \mathcal{G}_i$ moves away from the bounding box of $\mathcal{G}_i$. Taking b_i^x into the computation of α, we obtain an updated mixing weight $\overline{\alpha}_i^x = \frac{z_i^{pres}\pi_i^x b_i^x}{\sum_{j=1}^n z_j^{pres}\pi_j^x b_j^x} z_i^{pres}\pi_i^x b_i^x$. By employing such a boundary loss, the gradual exclusion of points from glimpses will be reflected in gradients to counter over-segmentation.

4 Experiments

4.1 Simulated Datasets

Dataset Generation. While many benchmark datasets have been established [23,25] for unsupervised object-centric learning, they do not come in the form of a point cloud. Thus, we introduce two new point-cloud datasets *Unity Object Room* and *Unity Object Table* built on the *Unity* platform [24]. The Unity Object Room (UOR) dataset is built to approximate the Object Room [25] dataset but with increased scope and complexity. In each scene, objects sampled from a list of 8 regular geometries are randomly placed on a square floor. The Unity Object Table (UOT) dataset approximates the Robotic Object Grasping scenario where multiple objects are placed on a round table. We populate each scene with objects from a pool of 9 objects with challenging irregular structures. For both datasets, the number of objects placed in each scene varies from 2 to 5 with

equal probabilities. During the scene generation, the size and orientation of the objects are varied randomly within a pre-defined range.

We capture the depth, RGB, normal frames, and pixel-wise semantics as well as instance labels for each scene from 10 different viewpoints. This setup aims to approximate the scenario where a robot equipped with depth and RGB sensors navigates around target objects. The point cloud data for each scene is then constructed by merging these 10 depth maps. For each dataset, we collect $50K$ training scenes, $10K$ validation scenes and $5K$ testing scenes.

Baseline. Due to the sparse literature on unsupervised 3D point cloud object-centric learning, we could not find a generative baseline to compare with. Thus, we compare SPAIR3D with PointGroup (PG) [22], a recent supervised 3D point cloud segmentation model. PointGroup is trained with ground-truth semantic labels and instance labels and performs semantic prediction and instance predictions on a point cloud. To ensure a fair comparison, we assign each point the same color (white). The PointGroup network is fine-tuned on the validation set to achieve the best performance.

Performance Metric. For UOR and UOT datasets, we use the Adjust Rand Index (ARI) [20] to measure the segmentation performance against the ground truth instance labels. We also employ foreground Segmentation Covering (SC) and foreground unweighted mean Segmentation Covering (mSC) [13] for performance measurements as ARI does not penalize object over-segmentation [13].

UOR UOT	PG	Ours	voxel size $0.75l$ voxel size $1.25l$	$6-12$ objects	object matrix
ARI↑	0.976	0.915 ± 0.03	0.932	0.912	0.872
	0.923	0.901 ± 0.02	0.922	0.892	0.879
SC↑	0.907	0.832 ± 0.04	0.853	0.846	0.856
	0.917	0.835 ± 0.03	0.857	0.843	0.877
mSC↑	0.900	0.836 ± 0.04	0.850	0.842	0.861
	0.907	0.831 ± 0.03	0.861	0.834	0.886

Table 1: 3D point cloud segmentation results on UOR (blue) and UOT (red).

Evaluation. Table 1 shows that SPAIR3D achieves comparable performance to the supervised baseline on both UOT and UOR datasets. As demonstrated in Fig. 4, each foreground object is proposed by one and only one glimpse. The scene layout is separated from objects and accurately modelled by the global VAE. It is worth noting that the segmentation errors mainly happen at the bottom of objects. Without appearance information, points at the bottom of objects are also correlated to the ground. In Fig. 2, we sort the test data based on their performance in ascending order and plot the performance distributions. As expected, the supervised baseline (Orange) performs better but SPAIR3D manages to achieve high-quality segmentation (SC score > 0.8) on around 80% of the scenes without supervision. The reported quantitative (Table 1) and qual-

itative results (Fig. 5(a)–(d)) show that our method achieves stable performance for those challenging scenes.

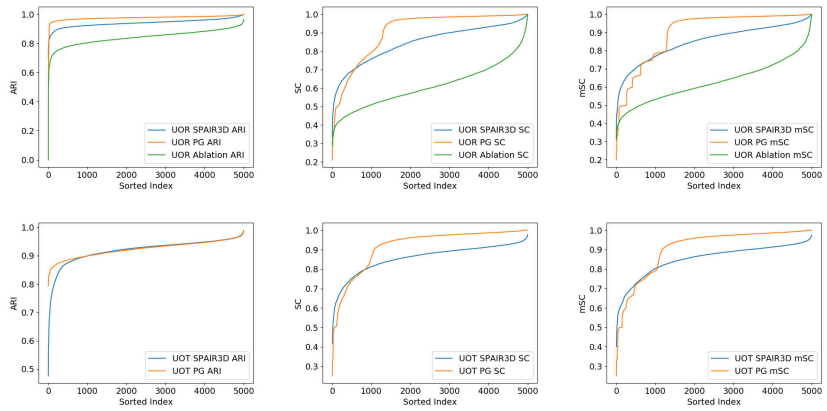


Fig. 2: Performance distributions on UOR (row one) and UOT (row two).

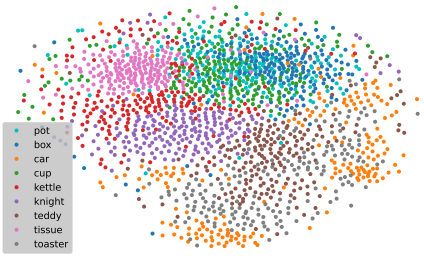
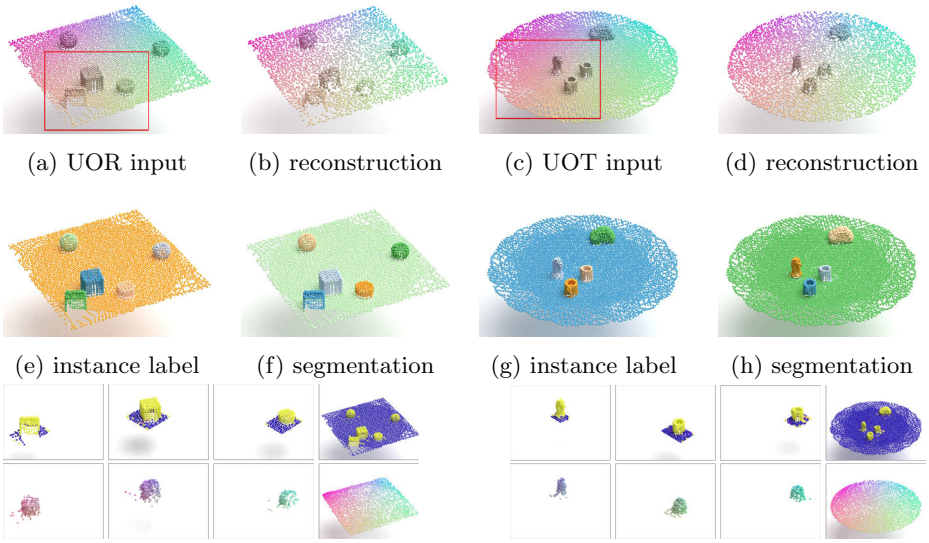


Fig. 3: t-SNE visualization of z^{what} on UOT

Object Centric Representation One advantage of our model is simultaneous segmentation and representation learning. To show that our model learns meaningful representations, we collect the z^{what} of 200 instances per-type and visualize them with the t-SNE algorithm [33]. Fig 3 shows the z^{what} of different object types occupy different regions. Note the embeddings of *pot* and *box* instances occupy the same area since they have almost identical spatial structure.

Voxel Size Robustness and Scalability In the literature, the cell voxel size, an important hyperparameter, is chosen to match the object size in the scene [10,30]. To evaluate the robustness of our method w.r.t voxel size, we train our model on the UOR dataset with voxel size set to $0.75l$ and $1.25l$ with l being the average size of the objects. Results in Table 1 show that our method achieves stable performance w.r.t the voxel size. To demonstrate scalability, we evaluate our pre-trained model on 1000 scenes containing $6-12$ randomly selected objects and report performance in Table 1. Due to the spatial invariance property, SPAIR3D suffers no performance drop on $6-12$ object scenes.



(a) UOR input (b) reconstruction (c) UOT input (d) reconstruction

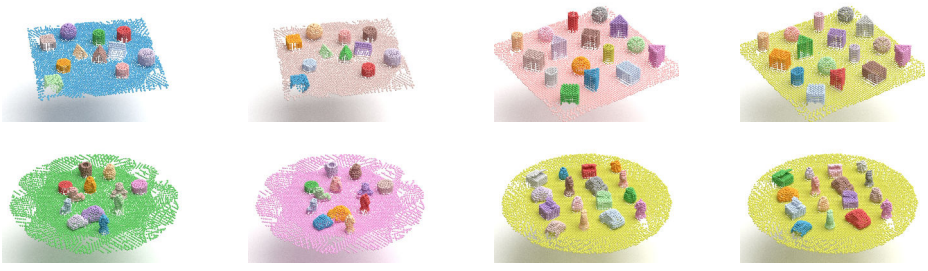
(e) instance label (f) segmentation (g) instance label (h) segmentation

(i) UOR close-up glimpses visualization, foreground alpha and scene layout reconstruction.

(j) UOT close-up glimpses visualization, foreground alpha and scene layout reconstruction.

Fig. 4: Visualization of segmentation results on UOR and UOT dataset.

We also evaluated our approach on scenes termed as *Object Matrix*, which consists of 16 objects placed in a matrix form. We fixed the position of all 16 objects but set their size and rotation randomly. For each dataset, SPAIR3D is evaluated on 100 *Object Matrix* scenes. The results are reported in Table 1. Note that our model is trained on scenes with 2 to 5 objects which is less than one-third of the number of objects in *Object Matrix* scenes. Fig 5(c)-(d) is illustrative of the results.



(a) instance label (b) segmentation (c) instance label (d) segmentation

Fig. 5: Segmentation on 6 - 12 object scenes (a,b) and object matrix (c,d)

4.2 Real Dataset

To demonstrate the performance of our approach on real data, we apply our model to the S3DIS [3] dataset, which contains point clouds of 6 large-scale indoor areas with 271 rooms (scenes).

Data Preprocessing. While the dataset contains objects from 13 semantic categories, we focus on objects with regular structures, including chairs, tables, and sofas. As our approach focuses on object-centric learning, we thus manually inspect the dataset and remove rooms that are too empty (such as hallways), containing clutter (such as storage rooms), and connected objects (such as lecture theater with connected chairs). Finally, we kept 174 scenes in total. We then downsample the dense point cloud of each scene for computational efficiency.

Baseline. Besides Point Group [22] as a supervised baseline, we use Mean-shift as a rule-based as well as an unsupervised baseline. Floors are removed before applying Mean-shift and the bandwidth parameter is determined by grid-search.

Performance Metric. Due to the scene diversity, instead of reporting the average per-scene ARI, SC, or mSC, we report the per-class mIoU on test sets. For our model and Point Group, we perform 6-fold cross-validation on the 6 areas and report the average.

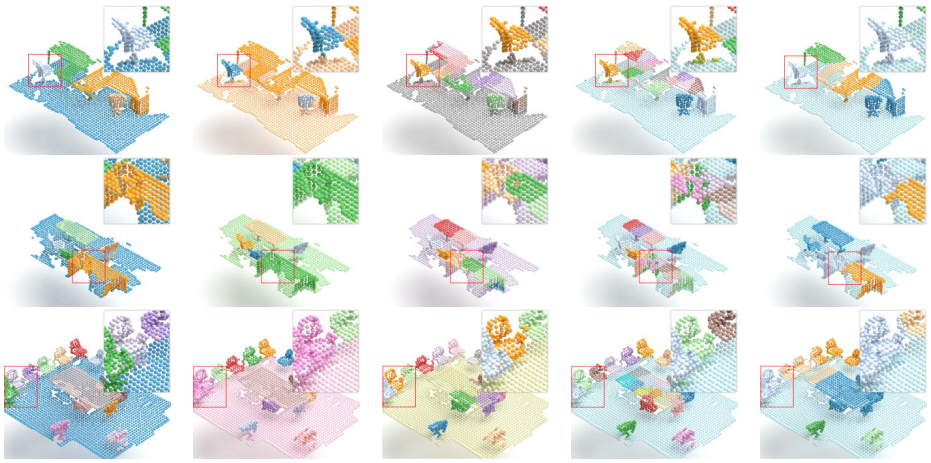
		Chair ↑	Table ↑	Sofa ↑	macro-avg ↑
PG	S	0.61	0.69	0.52	0.60
MS 0.06	U	0.75	0.34	0.36	0.48
MS 0.15	U	0.33	0.46	0.38	0.39
Ours	U	0.59	0.43	0.49	0.50

Table 2: Segmentation results on S3DIS. 'S' and 'U' denote the corresponding models are trained in supervised and unsupervised manner, respectively.

Evaluation. Per-class mIoU are reported in Table 2. Since the number of objects of different types varies greatly, we thus additionally report macro-average to show the overall performance of different models. Not surprisingly, Point Group (PG) achieves the highest mIoU across all categories. Similar to our model, Point Group misclassifies object points that are close to the floor as floor category (row 1 Fig. 6). Mean-shift (MS) does not share the same vulnerability since floors are manually detected and removed. However, Mean-shift is sensitive to the bandwidth value. The bandwidth reflects the prior object sizes. With the bandwidth tuned for Chair (MS 0.06) whose sizes are small on average, tables and sofas are largely over-segmented. With the bandwidth tuned for Table (MS 0.15), chairs tend to be under-segmented.

As demonstrated in row 3 Fig. 6, chairs are successfully segmented by our model even when multiple chairs are clustered together. The segmentation of tables presents a challenge for our model since their sizes are commonly larger than our maximum glimpses sizes. However, SPAIR3D still tries to expand the

glimpses sizes to better model larger objects while keeping the total number of glimpses low (row 2 Fig. 6). The experimental results in Table 2 demonstrate the potential of applying a generative model to more complicated scenarios.



(a) GT Label (b) PG (c) Ours (d) MS 0.06 (e) MS 0.15

Fig. 6: S3DIS segmentation results.

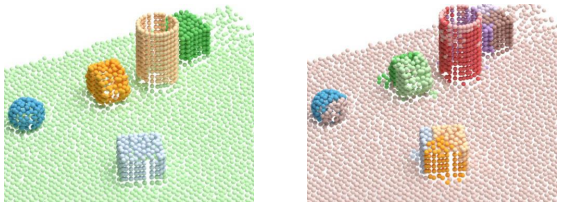
4.3 Ablation Study of Multi-layer PointGNN

To evaluate the importance of multi-layer PointGNN in $\mathbf{z}_i^{pres}$ generation (right branch in Fig. 1b), we remove the multi-layer PointGNN and generate $\mathbf{z}_i^{pres}$ directly from $\mathbf{f}_i$. The ablated model on the UOR dataset achieves **ARI:0.841**, **SC:0.610**, and **mSC:0.627**, which is significantly worse than the full SPAIR3D model. The performance distribution of ablated SPAIR3D (Fig 2, first row) indicates that removing the multi-layer PointGNN has a negative influence on the entire dataset. Fig. 7 shows that the multi-layer PointGNN is crucial to preventing over-segmentation.

4.4 Empirical Evaluation of PGD

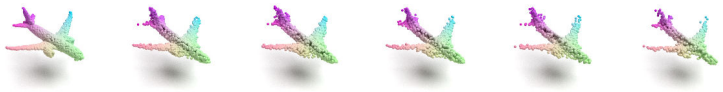
3D objects of the same category can be modeled by a varying number of points. The generation quality of the point cloud largely depends on the robustness of our model against the number of points representing each object. To demonstrate that PGD can reconstruct each object with a dynamic number of points, we train the global VAE on the ShapeNet dataset [7], where each object is composed of roughly 2000 points, and reconstruct the object with a varying number of points. For reference input point clouds of size N, we force PGD to reconstruct a point

cloud of size $1.5N$, $1.25N$, N, $0.75N$, and $0.5N$, respectively. As shown in Fig. 8, while with fewer details compared to the input, the reconstructions capture the overall object structure in all 5 settings.



(a) With PointGNNs (b) w/o PointGNNs

Fig. 7: The comparison between models with (left) and without (right) multi-layer PointGNNs. It shows that objects are over-segmented severely without multi-layer PointGNNs.



(a) (b) (c) (d) (e) (f)

Fig. 8: PGD trained on ShapeNet. (a) Input point cloud with N points. Reconstruction with (b) $1.5N$, (c) $1.25N$, (d) N, (e) $0.75N$, and (f) $0.5N$ points.

5 Limitations

Similar to SPAIR, the good scalability in SPAIR3D stems from the local attention and reconstruction mechanisms. By design, each voxel cell can only propose one object. Thus, it is difficult to detect multiple objects that exist in the same voxel cell. If one object is much larger than the size of the voxel cells, no voxel cells can accurately infer complete object information from its local perceptive field. One can alleviate the problem with overlapping voxel cells and make the mixture model hierarchical, which we leave as future work.

6 Conclusion and Future Work

Our proposed *SPAIR3D* algorithm is, to the best of our knowledge, the first generative unsupervised object-centric learning model on point cloud with applications to 3D object segmentation task. The experimental results demonstrate that SPAIR3D can generalize well to previously unseen scenes with a large number of objects without performance degeneration. The spatial mixture interpretation of SPAIR3D opens up the possibility to other extensions including memory mechanism [4] or iterative refinement [16], which is left as our future work.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: ICML. pp. 40–49. PMLR (2018)
2. Alemi, A., Fischer, I., Dillon, J., Murphy, K.: Deep variational information bottleneck. In: ICLR (2017), https://arxiv.org/abs/1612.00410
3. Armeni, I., Sax, A., Zamir, A.R., Savarese, S.: Joint 2D-3D-Semantic Data for Indoor Scene Understanding. ArXiv e-prints (Feb 2017)
4. Bornschein, J., Mnih, A., Zoran, D., Rezende, D.J.: Variational memory addressing in generative models. In: NIPS (2017)
5. Burgess, C., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., Lerchner, A.: Monet: Unsupervised scene decomposition and representation. ArXiv **abs/1901.11390** (01 2019), https://arxiv.org/abs/1901.11390
6. Burgess, C.P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., Lerchner, A.: Understanding disentangling in beta-vae. ArXiv **abs/1804.03599** (2018)
7. 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., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015)
8. Chang, B.M., Ullman, T., Torralba, A., Tenenbaum, B.J.: A compositional object-based approach to learning physical dynamics. ICLR (2017)
9. Chen, C., Deng, F., Ahn, S.: Roots: Object-centric representation and rendering of 3d scenes (2021)
10. Crawford, E., Pineau, J.: Spatially invariant unsupervised object detection with convolutional neural networks. AAAI **33**, 3412–3420 (07 2019). https://doi.org/10.1609/aaai.v33i01.33013412
11. Crawford, E., Pineau, J.: Exploiting spatial invariance for scalable unsupervised object tracking. AAAI **34**, 3684–3692 (04 2020). https://doi.org/10.1609/aaai.v34i04.5777
12. Diuk, C., Cohen, A., Littman, M.L.: An object-oriented representation for efficient reinforcement learning. In: ICML. p. 240–247. ICML '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1390156.1390187, https://doi.org/10.1145/1390156.1390187
13. Engelcke, M., Kosiorek, A.R., Jones, O.P., Posner, I.: Genesis: Generative scene inference and sampling with object-centric latent representations. In: ICLR (2020), https://openreview.net/forum?id=BkxfaTVFwH
14. Eslami, S.M.A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., Hinton, G.E.: Attend, infer, repeat: Fast scene understanding with generative models. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. p. 3233–3241. NIPS'16, Curran Associates Inc., Red Hook, NY, USA (2016)
15. Gadelha, M., Wang, R., Maji, S.: Multiresolution tree networks for 3d point cloud processing. In: ECCV. pp. 103–118 (2018)
16. Greff, K., Kaufman, R.L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M.M., Lerchner, A.: Multi-object representation learning with iterative variational inference. In: ICML (2019)
17. Greff, K., van Steenkiste, S., Schmidhuber, J.: Neural expectation maximization. In: NeurIPS. p. 6694–6704. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)

18. Henderson, P., Lampert, C.H.: Unsupervised object-centric video generation and decomposition in 3D. In: NeurIPS (2020)
19. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M.M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: ICLR (2017)
20. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification **2**, 193–218 (1985)
21. Islam*, M.A., Jia*, S., Bruce, N.D.B.: How much position information do convolutional neural networks encode? In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=rJeB36NKvB
22. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.W., Jia, J.: Pointgroup: Dual-set point grouping for 3d instance segmentation. CVPR (2020)
23. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. pp. 1988–1997 (07 2017). https://doi.org/10.1109/CVPR.2017.215
24. Juliani, A., Berges, V., Teng, E., Cohen, A., Harper, J., Elion, C., Goy, C., Gao, Y., Henry, H., Mattar, M., Lange, D.: Unity: A general platform for intelligent agents. ArXiv **abs/1809.02627** (2020)
25. Kabra, R., Burgess, C., Matthey, L., Kaufman, R.L., Greff, K., Reynolds, M., Lerchner, A.: Multi-object datasets. https://github.com/deepmind/multi-object-datasets/ (2019)
26. Kahneman, D., Treisman, A., Gibbs, B.J.: The reviewing of object files: Object-specific integration of information. Cognitive Psychology **24**(2), 175 – 219 (1992). https://doi.org/https://doi.org/10.1016/0010-0285(92)90007-O, http://www.sciencedirect.com/science/article/pii/001002859290007O
27. Kansky, K., Silver, T., Mély, D.A., Eldawy, M., Lázaro-Gredilla, M., Lou, X., Dorfman, N., Sidor, S., Phoenix, D.S., George, D.: Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. ArXiv **abs/1706.04317** (2017)
28. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014)
29. Li, N., Eastwood, C., Fisher, R.: Learning object-centric representations of multi-object scenes from multiple views. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 5656–5666. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/3d9dabe52805a1ea21864b09f3397593-Paper.pdf
30. Lin, Z., Wu, Y.F., Peri, S.V., Sun, W., Singh, G., Deng, F., Jiang, J., Ahn, S.: Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In: ICLR (2020), https://openreview.net/forum?id=rkl03ySYDH
31. Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. In: NeurIPS (2020)
32. Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2021)
33. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008), http://www.jmlr.org/papers/v9/vandermaaten08a.html
34. Shi, W., Rajkumar, R.: Point-gnn: Graph neural network for 3d object detection in a point cloud. In: CVPR. pp. 1708–1716 (2020)

35. Stelzner, K., Kersting, K., Kosiorek, A.R.: Decomposing 3d scenes into objects via unsupervised volume segmentation (2021)
36. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. In: Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing. pp. 368–377 (1999), https://arxiv.org/abs/physics/0004057
37. Wu, W., Qi, Z., Li, F.: Pointconv: Deep convolutional networks on 3d point clouds. In: CVPR. pp. 9613–9622 (06 2019). https://doi.org/10.1109/CVPR.2019.00985
38. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. pp. 4540–4549 (10 2019). https://doi.org/10.1109/ICCV.2019.00464