U3DS\textsuperscript{3}: Unsupervised 3D Semantic Scene Segmentation

Jiaxu Liu\textsuperscript{1} Zhengdi Yu\textsuperscript{1} Toby P. Breckon\textsuperscript{1,2} Hubert P. H. Shum\textsuperscript{1}
Department of \{Computer Science\textsuperscript{1} | Engineering\textsuperscript{2}\}, Durham University, UK
\{jiaxu.liu, zhengdi.yu, toby.breckon, hubert.shum\}@durham.ac.uk

Abstract

Contemporary point cloud segmentation approaches largely rely on richly annotated 3D training data. However, it is both time-consuming and challenging to obtain consistently accurate annotations for such 3D scene data. Moreover, there is still a lack of investigation into fully unsupervised scene segmentation for point clouds, especially for holistic 3D scenes. This paper presents U3DS\textsuperscript{3}, as a step towards completely unsupervised point cloud segmentation for any holistic 3D scenes. To achieve this, U3DS\textsuperscript{3} leverages a generalized unsupervised segmentation method for both object and background across both indoor and outdoor static 3D point clouds with no requirement for model pre-training, by leveraging only the inherent information of the point cloud to achieve full 3D scene segmentation. The initial step of our proposed approach involves generating superpoints based on the geometric characteristics of each scene. Subsequently, it undergoes a learning process through a spatial clustering-based methodology, followed by iterative training using pseudo-labels generated in accordance with the cluster centroids. Moreover, by leveraging the invariance and equivariance of the volumetric representations, we apply the geometric transformation on voxelized features to provide two sets of descriptors for robust representation learning. Finally, our evaluation provides state-of-the-art results on the ScanNet and SemanticKITTI, and competitive results on the S3DIS, benchmark datasets.

1. Introduction

As a crucial task in 3D computer vision, there has been increasing attention paid to point cloud segmentation in recent years due to its broad applicability to many real-world applications such as autonomous driving, virtual reality, robotics, and human-computer interaction. However, owing to the unordered and unstructured nature of point clouds, it is a non-trivial exercise to undertake segmentation upon them. In recent years, supervised point cloud segmentation approaches have made significant progress [1–7] against several benchmark datasets [8–11]. However, these approaches rely heavily on copious fully-annotated training data, in the form of labeled 3D point clouds. It is both time-consuming and labour-intensive to obtain such annotations accurately and consistently - especially for dense and complex 3D scenes. An alternative body of work leverages semi-supervised [12] and weakly-supervised [13–16] approaches to mitigate the labelled data requirements, but still require labour-intensive annotation at some level and lack of being readily scalable and adaptable to new datasets.

Our work aims to characterize 3D features without any explicit guidance allowing it to learn from the intrinsic structure of the data, and offer independence from erroneous, bias or inconsistent annotations, which significantly differ from prior weakly-supervised methods. To date, there are only a handful of prior works trying to address fully unsupervised segmentation for point clouds [17–20]. However, these approaches essentially focus on object-level segmentation or co-segmentation and cannot recover the full 3D scene labels without extra scene priors [18–20] and only a recent work [17] has attempted to address fully unsupervised semantic segmentation for 3D scenes. Our proposed new U3DS\textsuperscript{3} approach performs full holistic segmentation for the entire 3D scene in a scene-agnostic manner, spanning both indoor and outdoor scenarios across differing metric scales and achieving superior results on ScanNet [9] and SemanticKITTI [11] when compared to [17].

Despite the growth of unsupervised learning on 2D image segmentation [21–25], there is a lack of in-depth investigation into any 3D point cloud equivalent. Although some achievements in unsupervised segmentation learning have addressed 3D point cloud data via domain adaptation [26, 27], our work does not rely upon transfer learning. OGC [19] leverages the dynamic motion pattern of a
several variant feature representations from the same data
subsequent voxelized reverse geometric transformation on
volumetric transformations on the point cloud input and a
volumetric representations, we are able to apply differing
Finally, by leveraging the invariance and equivariance of the
classifier that operates solely on the feature space distance.
labels to provide new network parameters for the next itera-
tween features and clustering centroids to assign labels for
k
method by applying mini-batch
the superpoints. Following this, we propose the baseline
and merge them according to the distance and normals of
datasets, exhibits weaknesses in handling uneven data distri-
and struggles to form coherent clusters in the presence of outliers and data noise. Interestingly, some ex-
isting unsupervised approaches [17, 21] also incorporate k-
means as a component of their algorithms. On the other
hand, DBSCAN [29] encounters challenges when dealing
with categorical features, often fails to identify clusters
with varying densities, requires a drop in density to iden-
tify boundaries, and experiences decreased performance in
high-dimensional scenarios.

The goal of our approach is to enable a generalized method that is able to perform semantic segmentation for large-scale indoor and outdoor 3D scenes without utilizing any human labels or dynamic information between LiDAR frames. This paper takes a new step towards scene-level unsupervised semantic segmentation with a novel strategy. Specifically, we first apply voxel cloud connectivity seg-
mation (VCCS) [30] to generate the initial superpoint and merge them according to the distance and normals of
the superpoints. Following this, we propose the baseline method by applying mini-batch k-means [31] on
the features of a 3D point cloud to generate and update the clus-
tering centroids, and subsequently calculate the distance be-
tween features and clustering centroids to assign labels for
each point as pseudo-labels under the guidance of the super-
point. After that, we train the network with the pseudo-
labels to provide new network parameters for the next itera-
tion of clustering. Subsequently, we apply a non-parametric
classifier that operates solely on the feature space distance.
Finally, by leveraging the invariance and equivariance of the
volumetric representations, we are able to apply differing
volumetric transformations on the point cloud input and a
subsequent voxelized reverse geometric transformation on
these feature representations.

In this manner, our network is capable of producing several variant feature representations from the same data
source. This transformation operation is derived from a very
intuitive sense that the same inputs should result in similar predictions even under geometric transformation due to the
principle of invariance. Fundamentally, we learn a feature
representation that maximizes the effective semantic class separation. We provide two pathways to enforce color in-
variance and geometric equivariance that each provide our
underlying inductive bias for semantic consistency and geo-
metric structure by way of consistent clustering assignment
across the two pathways. This is performed via iterative
optimization of the clustering loss, which enforces a dis-
criminative feature space capable of high-level visual sim-
ilarity disambiguation. Finally, we train our voxel-based
method in an end-to-end manner. Furthermore, our evaluation
illustrates promising results across both indoor and out-
door datasets, S3DIS [8], ScanNet [9] and SemanticKITTI
[11], demonstrating the effectiveness and practicality of our
method and providing an initial reference performance for
completely unsupervised 3D semantic scene segmentation.
Overall, we propose a simple yet effective framework that
makes the new approach towards the task of unsupervised
point cloud segmentation for holistic 3D scenes, named
U3DS3. Fig. 1 illustrates an initial qualitative result of our
approach. Our key contributions are summarized as:

• We propose a novel unsupervised semantic segmenta-
tion method to leverage the invariance and equivari-
ance through geometric transformation for both 3D in-
door and outdoor holistic scenes.
• We analyze and compare existing clustering ap-
proaches and the concurrent state-of-the-art, demon-
strating the advantages and superiority of our method
for efficient unsupervised learning on large-scale point
clouds of holistic 3D scenes with faster convergence.
• We conduct extensive experiments and ablation studies
to demonstrate significant improvement over standard
baselines, across the S3DIS [8] ScanNet [9] and Se-
both the practicability of the proposed framework and
justify the intuition behind our design.

2. Related Work

In this section, we briefly summarize the prior literature
on 3D Semantic Segmentation (Sec. 2.1) and Unsupervised
Segmentation Learning (Sec. 2.2).

2.1. 3D Semantic Segmentation

To learn per-point semantics for 3D point clouds, many
deep learning based approaches tackle 3D point cloud se-
manetic segmentation tasks. PointNet [1] is a pioneering
work and the first one to leverage the point-based encoding
strategy, which is able to directly learn point features
from the raw points and extract local information embed-
ded in the neighbouring points. Following this work, more
point-based methods [2–4] have been proposed. KPConv
[3] designs a kernel function for operating convolution in 3D space to tackle local geometric structures. RandLA-Net [4] proposes a more efficient framework by replacing a complicated point selection strategy with random sampling. On the other hand, voxel-based approaches [32, 33] typically employ a 3D convolutional neural network by converting the point cloud from uneven distribution to regular voxel grids. Some works [34, 35] explore the more efficient voxel-based method by conducting sparse convolution. PVCNN [36] proposes a point-voxel corporate method and utilizes trilinear de-voxelization for voxelized features for fine-grained feature extraction. Our method also leverages a 3D convolutional neural network and follows the trilinear de-voxelization approach from [36] to avoid extracting identical features for the points that lie in the same voxel grid. Moreover, [37, 38] bring in graph convolutions to learn point features. However, all of these fully supervised methods require richly annotated training data, which are time-consuming and labour-expensive to obtain. To address the issue, Jiang et al. [12] propose a semi-supervised contrastive learning method for alleviating the tedious labelling cost. Zhang et al. [13] utilizes perturbed self-distillation to employ a weakly supervised method for point cloud semantic segmentation and reducing human annotations. In terms of unsupervised manner, KMeans [28] and DBSCAN [29] are classic methods and have no requirement for labelled data. However, these methods can only deal with simple object-level segmentation and lack of robustness under non-convex and uneven data distribution.

2.2. Unsupervised Segmentation Learning

The exploration of unsupervised 2D semantic segmentation shows more maturity compared to 3D. DeepCluster [39] clusters the feature vector of the entire dataset using k-means [28] to assign pseudo-labels and subsequently trains its encoder. Our method shares the common idea that iterative optimization of clustering can improve feature representation learning. Abdal et al. [25] propose an unsupervised segmentation framework that enables foreground and background separation for raw input images and segments common 3D objects. Some other works [26, 27] focus on unsupervised domain adaptation for point cloud semantic segmentation. PICIE [21] notably proposes a pixel-level semantic segmentation method that can incorporate geometric consistency as an inductive bias to learn invariance and equivariance for photometric and geometric variations.

As for the 3D domain, some work already shows notable performance on unsupervised segmentation, but not especially for scene-level semantic segmentation. Yang et al. [18] employs object sampler and background sampler to tackle unsupervised point cloud co-segmentation for object-level segmentation by co-contrastive learning and mutual attention sampling. However, the co-segmentation method only works for objects and it is limited by the groups of common 3D objects. Some other works [26, 27] focus on unsupervised domain adaptation for point cloud semantic segmentation. OGC [19] can simultaneously identify multiple 3D objects in a single forward pass, without any human annotations, which leverages the dynamic motion patterns over (LiDAR captured) sequential point clouds as supervision signals to automatically discover rigid objects. However, OGC [19] needs the dynamic information of continuous point cloud frames as an input prior. Poux et al. [41] leverages the region growing method for indoor...
unsupervised object-level segmentation, but the segmentation is only for generating larger object segment parts. GrowSP [17] is the only unsupervised 3D semantic segmentation that employs a progressively region-growing scheme to generate high-quality over-segmentation, however, their method does not fully leverage the intrinsic geometric information of the point clouds and tends to predict over-smooth segmentation and lose accuracy in intricate scenes. In contrast, our work aims to investigate unsupervised 3D semantic segmentation leveraging the intrinsic geometric information of the point clouds for holistic 3D scenes without any dynamic information or transfer learning prior.

3. U3DS$^3$ Methodology

This work formulates the task of unsupervised point cloud semantic segmentation as point-level segmentation, where every point within the point cloud needs to be assigned a label of a fixed number of semantic class labels.

To state formally, given a point cloud set $P$ without labels, let $c = \{c_i\}$ and $f = \{f_i\}$ denote the point coordinates and the corresponding features from $P \in \mathbb{R}^{N \times 3}$, $F \in \mathbb{R}^{N \times d}$, where $N$ is the number of points of the input point cloud, and $d$ denotes the feature size, which contains coordinates, colours, and normalized positional information. Hence, the goal of this work is to learn a semantic segmentation function $g_{\theta}$, which is able to predict per-point labels in an unsupervised way for $P$ using only $c$ and $f$.

As shown in Fig. 2, for each input point region, we first apply two different colour transformations and afterwards convert them to the volumetric domain. For pathway-1 in the top row, we implement a geometric transformation before the voxelized features are fed into the model. After the forward pass, we operate a corresponding inverse geometric transformation to the output features to ensure this representation shares the same properties with the non-transformed pathway-2. Subsequently, we cluster features from the different point cloud blocks and produce two groups of clustering centroids and labels, which can be used for further training and loss assembled from different pathways.

3.1. Superpoint

For all point clouds $P_1, P_2, P_3, \ldots$ within a point cloud set $P$, we adhere to the VCCS [30] method to obtain initial superpoints for each point cloud. These can be denoted as $\{SP_1^1, SP_1^2, SP_1^3, \ldots\}, \{SP_2^1, SP_2^2, SP_2^3, \ldots\}, \ldots$, where $SP_i^j$ represents the $j$-th superpoint in the $i$-th point cloud. The initial superpoints may vary across different point clouds. Subsequently, we employ a straightforward strategy to merge the superpoints within each scene: 1) Identify the smallest superpoint $SP_i^1$ along with its two closest neighboring superpoints $SP_i^{k_1}, SP_i^{k_2}$; 2) Compute the vector addition of points within each superpoint and calculate the cosine similarity, here simply noted as $\cos[SP_i^j, SP_i^{k_1}]$; 3) Merge the smallest superpoint with the one that exhibits higher cosine similarity; 4) Repeatedly execute the above three steps until the superpoints reach a predetermined number. This simplistic approach is based on the principle that similar semantic objects possess comparable normals. Ultimately, the updated superpoints become $\{\{SP_1^{i_1}, SP_1^{i_2}, \ldots\}, \{SP_2^{i_2}, SP_2^{i_3}, \ldots\}, \ldots\}$, ensuring that the points within the same superpoint are assigned identical labels. We define the final superpoint count as a parameter, represented by $\gamma_{sp}$. For all datasets, the optimal value is empirically found as $\gamma_{sp} = 40$.

3.2. Voxelization and Devoxelization

We produce different representations for the input point cloud via the geometric transformation on the volumetric domain, where a voxel-based architecture is naturally adopted for such representation. Here, using voxelization and devoxelization in the pipeline, we present a simple yet effective network which contains only 3D convolutional layers with batch normalization without any additional component (details in Sec. 3.3).

Given the input points coordinate $c$ with corresponding features $f$ in the input blocks, we normalize the coordinates $c$ before voxelizing the original points to gain scale invariance. Specifically, we normalize the coordinate $c$ into $[0,1]$ and denoted by $c^* = \{c_i^*\}$. In this process, the point features (including the coordinates) do not change, and the normalized coordinates are only used for converting the feature to the proper volumetric space.

When transferring the features $f$ with normalized coordinates $c^* = \{x^*, y^*, z^*\}$ into the voxel grids $\{V_{m,p,q}\}$, the interpolated feature $f_i$ for the voxel grid is calculated as the mean value of the features of points located in the grid.

$$V_{m,p,q} = \frac{1}{K_{m,p,q}} \sum_{i=1}^{n} I_{\text{floor}}(x_i^* \times r) = m,$$

$$I_{\text{floor}}(y_i^* \times r) = p, I_{\text{floor}}(z_i^* \times r) = q \times f_i,$$

where $r$ denotes the voxel resolution and $I$ is an indicator function that indicates whether coordinates $c_i$ belong to the voxel grid $\{m, p, q\}$. $K_{m,p,q}$ represents the count of points falling within the grid $\{m, p, q\}$, and $\text{floor}(\cdot)$ is floor function that outputs the greatest integer less than or equal to the input.

In terms of the per-point clustering, we need to devoxelize the voxel-based features output from the model $g_\theta$ to point-based features. We follow the trilinear interpolation of PVCNN [36] instead of the traditional nearest neighbor interpolation to ensure that nearby points are not assigned identical features.

3.3. Baseline: Clustering and Iteration

U3DS$^3$ applies a clustering-based method iteratively to generate pseudo-labels and train our baseline method, as in-
spired by DeepCluster [39]. Adapting [39] to the 3D domain is non-trivial due to the irregular nature and varying sparsity of point clouds. We present a simple yet effective strategy: switching between generating pseudo-labels via clustering with the current feature representations, and training new feature representations with the generated pseudo-labels. Different from [21,39], the segmentation function $g_0$ should be able to produce per-point features, and we replace the parametric classifier with a non-parametric distance metric. Specifically, we denote the voxelization and devoxelization operations as $Z$ and $Z^{-1}$. The voxelized feature is $v = \{ v_i \} = \{ Z(f_i, c_i^*) \}$, and the output voxelized feature of the 3D convolutional function is $v^{out} = g_0(v)$. Finally, the features for clustering can be denoted as $f = \{ f_i \} = \{ Z^{-1}(v^{out}_i, c_i^*) \}$. The main procedure can be separated as two parts:

1. Using the current embeddings and k-means to cluster the points with superpoints guidance in the point cloud:

   $$\min_{l,\mu} \sum_i \| f'_i - \mu_{l_i} \|^2 \quad (2)$$

where $l_i^p$ denotes the cluster label of point $c_i$ with the constraint of superpoint, and $\mu_k$ denotes the k-th cluster centroid. Note the features $f'_i$ and the centroids $\mu_k$ have the same dimension.

2. Using the class labels as pseudo-labels, we train a classifier via cross-entropy loss, which is shown in the point cloud setting as:

   $$\min_{\theta, W} \sum_i L_{CE} \left( g_W(f'_i), l_i^p, \mu \right) \quad (3)$$

where $g_W$ denotes the parametric classifier. Under the unsupervised setting, it will be very challenging to train a classifier jointly with constantly changing pseudo-labels. We therefore choose to label points only based on their cosine distance from to the clustering centroids in feature space. Specifically, the loss function shows the following format:

   $$\min_{\theta} \sum_i L_{cluster} \left( f'_i, l_i^p, \mu \right) \quad (4)$$

   $$L_{cluster} \left( f'_i, l_i^p, \mu \right) = -\log \left( \frac{e^{-d(f'_i, \mu_{l_i}^p)}}{\sum_e e^{-d(f'_i, \mu_e)}} \right) \quad (5)$$

where $d(\cdot, \cdot)$ denotes the cosine distance.

### 3.4. Volumetric Transformations

To improve robustness in the unsupervised setting under different scenarios, we leverage the invariance and equivariance of volumetric representations of point clouds. Invariance means that the labelling should not change after applying different transformations such as colour jittering. Equivariance in the volumetric domain means when we apply a geometric transformation to the point cloud, the corresponding 3D convolutional feature should be similarly transformed, and the corresponding labels are also wrapped according to this transformation.

For simplicity, we name the two pipelines processing the two representations as pathway-1 and pathway-2. To produce two different representations for an individual input block, we apply a geometric transformation before volumetric feature extraction and then perform a corresponding inverse transformation on the final voxelized features.

Specifically, let $G$ and $G^{-1}$ denote the voxelized feature geometric transformation and its reverse transformation respectively, and $O$ is the colour transformation. For point $c$ with its feature $f$, we apply different colour transformations for original features $f$:

$$f_1 = O_1(f), f_2 = O_2(f) \quad (6)$$

Next, we transform these two features into the voxel grid, noting that $c_2^*$ is actually equal to $c_2^*$:

$$v_1 = Z(f_1, c_1^*), v_2 = Z(f_2, c_2^*) \quad (7)$$

After that, the voxelized feature transformations are applied to the volumetric domain: only the features of pathway-1 are transformed whilst the other remains unchanged. The geometric transformations operate on the voxelized feature $v$ and the corresponding reverse geometric transformations operate on the output voxel feature $v^{out}$:

$$v^{out}_1 = G^{-1} \left\{ g_0(G(v_1)) \right\}, v^{out}_2 = g_0(v_2) \quad (8)$$

Subsequently, we perform de-voxelization to get the features for clustering:

$$f'_1 = Z^{-1}(v^{out}_1, c_1^*), f'_2 = Z^{-1}(v^{out}_2, c_2^*) \quad (9)$$

### 3.5. Losses and Labelling Scheme

Given input clouds $c$ with features $f$, according to the colour and geometric transformations introduced in Sec. 3.3, two different feature representations, $f'_1, f'_2$, can be produced. By leveraging these two features, we cluster the two representations separately to get two groups of centroids and pseudo-labels:

$$l^{(1)}(1), \mu^{(1)} = \arg \min_{l,\mu} \sum_i \| f'_{1i} - \mu_{l_i} \|^2 \quad (10)$$

$$l^{(2)}(2), \mu^{(2)} = \arg \min_{l,\mu} \sum_i \| f'_{2i} - \mu_{l_i} \|^2 \quad (11)$$
We then set two loss functions. Firstly, the feature representation should match the pseudo-labels produced by the same pathway:

$$L_1 = \sum_i L_{\text{cluster}} \left( f'_{i1}, l_{i1}^{p(1)}, \mu^{(1)} \right) + \sum_i L_{\text{cluster}} \left( f_{2i1}, l_{2i1}^{p(2)}, \mu^{(2)} \right)$$

(12)

Similarly, the feature representation should whilst match the pseudo-labels produced by the different pathway:

$$L_2 = \sum_i L_{\text{cluster}} \left( f'_{i2}, l_{i2}^{p(2)}, \mu^{(2)} \right) + \sum_i L_{\text{cluster}} \left( f_{2i2}, l_{2i2}^{p(1)}, \mu^{(1)} \right)$$

(13)

The final training objective is their summation:

$$L_{\text{final}} = L_1 + L_2$$

(14)

The loss encourages the feature from one pathway to adhere to labels generated by another pathway, which encourages the network to label similarly to feature representations from different pathways.

**Hungarian Algorithm:** To match the clustering labels with the real labels, we utilize the Hungarian algorithm [42] across the whole dataset every epoch. Specifically, where $C$ is categories, $P$ is the predicted set and $G$ is the ground truth (GT) set. $S^{C \times C}$ is the matching matrix, where $S_{ij}$ denotes the matching degree between $i^{th}$ predicted category and $j^{th}$ GT category. Criterion: finding bijection $f : i \rightarrow j$ to maximize $\sum_{i=1}^{C} S_{i,f(i)}$.

4. Experiments

**Implementation Details:** We implement a simple yet effective framework with 8 layers 3D convolution, where each layer employs a 3D batch normalization and leaky rectified linear activation function (ReLU). The input point cloud contains 12D features, i.e., the point coordinates $(x, y, z)$ in the normalized block coordinate system, colour information $(R, G, B)$, per-point normals and normalized raw coordinates in the original scene coordinate system. Note that no colour information is provided in SemanticKITTI [11].

**Training:** We use a batch size of 4 with 4096 points per batch for all datasets. The chosen optimizer is stochastic gradient descent (SGD) with a learning rate of $10^{-2}$ and a weight decay of $10^{-4}$. We train our network for 10 epochs. For the geometric transformation in the volumetric domain, we reverse the order of tensors along the given $x$, $y$ and $z$-axis respectively. The colour transformation comprises random contrast and random brightness adjustment. The output feature dimension from the model and the clustering feature dimension is set to 128. The resolution of the voxel grid is set to 32. Besides, we use the FAISS library [43] on GPU to compute the cluster centroids via employing a mini-batch $k$-means approach [31].

**Evaluation:** For evaluation and comparison with other methods, we choose two classical unsupervised clustering methods, $k$-means [28], DBSCAN [29], and the only unsupervised semantic segmentation method GrowSP [17] as baselines. Our method is evaluated with three metrics: overall accuracy (oAcc), mean accuracy (mAcc) and the mean intersection of union (mIoU) on all datasets. All experiments are performed on a single NVIDIA RTX 2080Ti GPU.

<table>
<thead>
<tr>
<th>Method</th>
<th>Level of Supervision</th>
<th>mIoU</th>
<th>mAcc</th>
<th>oAcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>KMeans [28]</td>
<td>unsupervised</td>
<td>3.4</td>
<td>10.4</td>
<td>10.2</td>
</tr>
<tr>
<td>DBSCAN [29]</td>
<td>unsupervised</td>
<td>6.1</td>
<td>10.1</td>
<td>15.3</td>
</tr>
<tr>
<td>GrowSP [17]</td>
<td>unsupervised</td>
<td>25.4</td>
<td>44.2</td>
<td>57.3</td>
</tr>
<tr>
<td>U3DS (ours)</td>
<td>unsupervised</td>
<td>27.3</td>
<td>46.8</td>
<td>60.1</td>
</tr>
</tbody>
</table>

Table 1. Semantic segmentation results on ScanNet dataset. We evaluate 20 categories on validation set.
Figure 4. Qualitative results on SemanticKITTI [11] (top 2 rows) and S3DIS [8] (bottom row). Our method draws more versatile results compared with DBSCAN [29] and is more stable than k-means [28], which shows promising segmentation results.

## 4.1. Datasets

We evaluate U3DS$^3$ on two indoor and one outdoor benchmark: S3DIS [8], ScanNet [9] and SemanticKITTI [11].

**S3DIS** [8] is a large-scale indoor scenes dataset which consists of 271 point cloud rooms in six areas. The annotations of each point in the point cloud scene belong to 13 semantic categories. We train the model in areas 1, 2, 3, 4, 6 and test it in area 5 following [1, 3, 44]. We exclude clutter and test with 12 classes for a fair comparison with GrowSP [17], nevertheless, we also test with 13 categories to compare with the existing supervised, weakly, and semi-supervised methods.

**ScanNet-v2** [9] is an RGB-D real-world indoor dataset. It contains 1201 scenes for training, 312 for validation, and 100 for online testing. For scene semantic segmentation, it has 40 classes and one unlabelled class for training and 20 classes and for testing. We compare with existing clustering and unsupervised methods on the validation set.

**SemanticKITTI** [11]: is a large-scale outdoor dataset that is based on the KITTI Vision Odometry Benchmark. For the semantic segmentation task, it provides 22 sequences with point-wise annotation of 19 classes. Each sequence contains a number of scene scans collected by the complete 360 field-of-view of the employed automotive LiDAR, where sequences 11-21 are used for online testing.

is the validation set and the others are training sets.

### Data Preparation:
For all datasets, we choose $\gamma_{sp} = 40$ as the superpoint number for each scene. We first apply uniform downsampling to S3DIS [8] and ScanNet [9] with the sub-grid size 0.03 and subsequently follow PointCNN [44] to sample point clouds into blocks to ensure that each data sample in the batch has the same number of points. For S3DIS [8] and ScanNet [9], the block size is $1.5 \times 1.5$ on $xy$ plane, and each block contains 4096 points. For SemanticKITTI [11], we set each block size as $5 \times 5$ on $xy$ plane with 4096 points. For each point cloud, we utilize VCCS [30] to derive the initial superpoint. This is then merged for enhanced segmentation, as detailed in Sec. 3.1. Furthermore, due to the characteristics and predominance of roads in outdoor SemanticKITTI [11] datasets, we apply RANSAC [45] to fit a plane as the road for improved generation of superpoints. Note this process will not be utilized elsewhere.

<table>
<thead>
<tr>
<th>Method</th>
<th>Level of Supervision</th>
<th>mIoU</th>
<th>mAcc</th>
<th>oAcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>KMeans [28]</td>
<td>unsupervised</td>
<td>2.5</td>
<td>8.1</td>
<td>8.2</td>
</tr>
<tr>
<td>DBSCAN [29]</td>
<td>unsupervised</td>
<td>6.8</td>
<td>7.5</td>
<td>17.8</td>
</tr>
<tr>
<td>GrowSP [17]</td>
<td>unsupervised</td>
<td>13.2</td>
<td>19.7</td>
<td>28.3</td>
</tr>
<tr>
<td>U3DS$^3$ (ours)</td>
<td>unsupervised</td>
<td>14.2</td>
<td>22.1</td>
<td>34.8</td>
</tr>
<tr>
<td>Table 2. Semantic segmentation results on S3DIS Area-5 are compared using mIoU, mAcc and oAcc across various methods. Where (12) indicates the exclusion of clutter, while the results without (12) are tested with 13 classes.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.2. Results and Comparison on Benchmarks

To thoroughly evaluate our U3DS\textsuperscript{3}, we test our methods on the indoor S3DIS [8], ScanNet [9] and outdoor SemanticKITTI [11] benchmarks. Tabs. 1 to 3 respectively shows the semantic segmentation results on the ScanNet, SemanticKITTI and S3DIS dataset. Not surprisingly, fully supervised methods provide the best performance. From Tab. 3, our method significantly outperforms the existing clustering methods, where it achieves 75.5\% overall accuracy and 42.8 mIoU on the S3DIS dataset. Moreover, our method is even close to the performance reported by the fully supervised method [1] and some up-to-date weakly supervised methods [48, 49], which is a big step forward for unsupervised semantic 3D scene segmentation.

Moreover, we outperform GrowSP [17] on both the ScanNet and SemanticKITTI datasets. Specifically, as displayed in Tab. 1, our method achieves a superiority of +1.9 mIoU and +2.6 mAcc over their results. Additionally, Tab. 2 demonstrates that our method achieves 1 mIoU and 3.4 mAcc higher than GrowSP [17], despite having a slightly lower oAcc. Fig. 3 shows the qualitative comparison on S3DIS, which further demonstrates the superiority of our method.

4.3. Ablation Study

To showcase the effectiveness of each module and the different volumetric transformations. We conduct eight groups of experiments on the S3DIS [8] dataset: (1) the baseline approach proposed in Sec. 3.3, (2) adding colour transformation on the basis of the control group (1), (3) adding voxelized feature transformation on the basis of the control group (1), and (4) full model without prior superpoint, (5)-(8) different final prior superpoints as guidance. As shown in Tab. 4, our full model clearly outperforms the baseline on all of the evaluation metrics, benefiting from the delicate volumetric transformation design and superpoint prior. Groups (3) and (4) outperform by +5 mIoU and 8 OA compared to the baseline. More interestingly, the improvement of adding the geometric transformation for equivariance is more significant than that of the invariance transformations, which is different from prior unsupervised learning work in the 2D domain [24, 25, 40]. It is known that point clouds essentially present much stronger geometric priors than 2D images with explicit 3D structures, which we believe can significantly help the 3D representations to be more robust and consistent cross-view and less sensitive to light changes and jittering. Moreover, the employment of superpoints can significantly enhance the overall performance. This enhancement is a result of the more abundant information of superpoints, which facilitates the pre-segmentation of the scene into higher-level semantic classes. Additional results are available in the supplementary material.

4.4. Analysis

Our U3DS\textsuperscript{3} approach demonstrates a promising level of performance on both indoor and outdoor datasets when compared to existing baselines. In contrast to GrowSP [17], our method achieves superior results on ScanNet [17] and SemanticKITTI [11]. As the scene complexity increases, the quality of GrowSP [17] superpoints tends to degrade. In contrast, our approach not only incorporates pre-segmentation but also employs a two-pathways training algorithm, leveraging the concepts of invariance and equivariance.

Nonetheless, slight performance degradation can occur in practical scenarios. To address this, we have implemented three strategies: (i) splitting the largest cluster when another cluster in the set reaches zero entities; (ii) applying mild centroid perturbation during updates; and (iii) re-weighting for loss balancing using per-class pseudo-label ratios at each epoch. Additionally, our two-pathways approach expedites the convergence time during training. For instance, while training with only one pathway necessitates around 8 epochs to achieve convergence, the two-pathways approach accomplishes convergence in just 2-3 epochs.

4.5. Conclusion and Discussion

We propose a novel generalized unsupervised semantic segmentation method for both indoor and outdoor 3D scenes with objects and the background. Our method leverages a simple yet effective framework via clustering and iterative generation leveraging the invariance and equivariance of the volumetric representations with the assistance of superpoint. Experiments show promising performance on S3DIS, ScanNet and SemanticKITTI datasets which proves the superiority of our approach beyond all the existing baselines. This work aims to provide more insight for 3D unsupervised learning. Future work will explore improved point sampling strategies and an extension to point- or graph-based representations, benefiting other areas related to unsupervised learning, metric learning and 3D representation learning.

Acknowledgement: EPSRC NortHFutures (ref: EP/031012/1).
References


[23] Yassine Ouali, Céline Hudelot, and Myriam Tami, “Autoregressive unsupervised image segmentation,” in *Eur...*