ABSTRACT

Learning on 3D scene-based point cloud has received extensive attention as its promising application in many fields, and well-annotated and multisource datasets can catalyze the development of those data-driven approaches. To facilitate the research of this area, we present a richly-annotated 3D point cloud dataset for multiple outdoor scene understanding tasks and also an effective learning framework for its hierarchical segmentation task. The dataset was generated via the photogrammetric processing on unmanned aerial vehicle (UAV) images of the National University of Singapore (NUS) campus, and has been point-wise annotated with both hierarchical and instance-based labels. Based on it, we formulate a hierarchical learning problem for 3D point cloud segmentation and propose a measurement evaluating consistency across various hierarchies. To solve this problem, a two-stage method including multi-task (MT) learning and hierarchical ensemble (HE) with consistency consideration is proposed. Experimental results demonstrate the superiority of the proposed method and potential advantages of our hierarchical annotations. In addition, we benchmark results of semantic and instance segmentation, which is accessible online at https://3d.dataset.site with the dataset and all source codes.

CCS CONCEPTS
• Computing methodologies → Scene understanding; Neural networks; 3D imaging.

KEYWORDS
Point cloud; scene understanding; hierarchical learning; semantic segmentation; instance segmentation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Xinke Li1, Chongshou Li1,∗, Zekun Tong1, Andrew Lim1, Junsong Yuan2, Yuwei Wu1, Jing Tang1, Raymond Huang1
1Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore
2Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY, USA
{xinke.li, zekuntong}@u.nus.edu, {iseul, isedlmm}@nus.edu.sg, j.s.yuan@buffalo.edu
ywwu@u.nus.edu, {isejtang, raymond.huang}@nus.edu.sg

1 INTRODUCTION

Due to the significant progress of 3D sensing technologies in recent years, multiple sources of 3D point cloud become affordable and easily acquired. Reconstruction of outdoor scene from point cloud has also received an increasing interest, which is critical for various areas such as urban planning and management [5], vehicle navigation [4], virtual reality [7] as well as simulation [21]. As the fundamental step of reconstruction, scene understanding with point cloud data can be greatly facilitated by recent advances of machine learning techniques especially the deep learning. Large and well-annotated datasets play a leading role for the successful application of these techniques.

Although dozens of 3D scene-based point cloud datasets are proposed [1, 3, 9, 10, 14, 25, 26, 28, 31], majority of them are not perfectly fit for outdoor scene reconstruction. Firstly, the datasets may face various limitations from their sources which are either RGB-D images [1, 9, 10, 28] or light detection and ranging (LiDAR) based mobile laser scanning (MLS) [3, 25, 26, 31] and terrestrial laser scanning (TLS) [14]. The RGB-D data can be easily obtained and processed via a mature pipeline [9], while it is likely prevented from capturing outdoor environment by the limited measurement range. The LiDAR scanner usually results in unavoidable severe occlusions and expensive equipment costs although it is good at capturing large-scale scenes [19]. Secondly, the annotations of extant datasets are not targeted for outdoor scene reconstruction. Following the

Figure 1: Overview of six regions of the Campus3D dataset.
well-established data format CityGML [17], a standard urban model should contain fine structures of building and other artifacts. However, such fineness is not presented by current annotations which mainly consist of indoor objects or traffic elements [3]. Thus, it is necessary to build new datasets with the aim of supporting scene understanding based automatic reconstruction.

In this work, we construct a photogrammetry point cloud dataset Campus3D from UAV imagery over the National University of Singapore (NUS) campus of 1.58 km² area. Due to the recent progress of Structure from Motion (SfM), Multi-View Stereo (MVS) and UAV techniques [11, 39], photogrammetry point cloud is easily accessible from unmanned aerial vehicle (UAV) imagery. This type of data source is able to fulfill the requirement of scene reconstruction because UAV imagery is robust to occlusion, and can effectively obtain the holistic view of the scene.

Inspired by the multiple levels of details (LoD) in CityGML [17] for the reconstruction, we point-wisely annotate this dataset with hierarchical multi-labels for both semantic and instance segmentation. For a data point, an example annotations is construction->building->wall/roof. The fine-grained label (e.g., wall/roof) can match the LoD2 for reconstruction [32], where the building model is detailed to roof and wall structures. For the further study on it, we organize the labels as a tree with five hierarchical (granularity) levels displayed by Figure 3. In the end, the whole dataset present a holistic view of scene, which contains 0.94 billion points with 2,530 modality-based instances, 24 semantic classes and 6 pattern-based regions as displayed by Figure 1.

The proposed dataset with hierarchical annotations is expected to promote better outdoor scene understanding. Based on the constructed label tree, we formulate a hierarchical learning (HL) problem for semantic segmentation, and propose a new metric for consistency across granularity levels named Consistency Rate (CR). Besides accuracy, prediction consistency is an important issue for the HL. For example, if one point is predicted as “roof” at fine-grained level, the results at the corresponding coarse level must be “building” and “construction”(see Figure 3), otherwise, it is a violation of the hierarchical relationship. Taking this into consideration, we introduce a two-stage method consist of multi-tasking (MT) learning and hierarchical ensemble (HE). The MT based on neural models jointly learns semantic labeling on different granularity levels. The post-processing HE rigidly ensures the results to fulfill the hierarchical consistency by choosing the most likely root-to-leaf path of the label tree. The results of CR and segmentation task suggest the goodness that the HL method utilizes the hierarchical relationship and the chance that hierarchical annotations assists segmentation tasks.

Furthermore, we establish the benchmarks on the dataset via applying deep models for two classic scene understanding tasks: (1) semantic segmentation and (2) instance segmentation. For the concern of computational efficiency and compatibility to point-based models, we investigate the data preprocess technique and two sampling methods: (1) random block sampling (RBS) and (2) random centered K nearest neighbor (RC-KNN) sampling. And the RBS is chosen as the unified sampling method for benchmarks in view of its better performance.

We summarize the contributions of this paper as follows:

• A photogrammetry point cloud dataset with hierarchical and instance-based annotations is present. Moreover, an accessible workflow of the acquisition and annotation is provided.
• An effective two-stage method for the formulated hierarchical semantic segmentation on point cloud is proposed. Experimental results demonstrate the superiority of our HL methods over the non-HL method in terms of both hierarchical consistency and segmentation performance.
• We propose new benchmarks for semantic segmentation and instance segmentation on 3D point cloud, and release the source codes of the training/evaluation framework as well as the dataset. These benchmarks are standardized with consideration of the unified data prepossess techniques and sampling methods.

2 RELATED WORK

In this section, we firstly review the existing 3D scene-based point cloud datasets and compare our dataset with them in detail below. Based on the application area, we briefly divide the existing datasets into two categories: (1) indoor scene datasets and (2) outdoor scene datasets. A summary of comparison between Campus3D and the widely-used datasets is provided by Table 1, and additional comparisons in terms of annotation are provided in the supplementary document. Secondly, we briefly review the existing deep neural models for point cloud segmentation.

Indoor Dataset. Indoor scene understanding is an active research area, and many datasets have been reported in literature [1, 2, 6, 9, 16, 27–29, 40]. These datasets are usually generated by RGB-D images which can be easily got by cheap sensors (e.g., Microsoft Kinect). Early datasets NYUv2 [28], SUN3D [40] and SUN RGB-D [29] were annotated by either polygons in 2D [28, 29, 40] or bounding box in 3D [29], of which the information for 3D scene reconstruction (e.g., semantic segmentation, surface reconstruction, meshes, etc.) is limited. Recently released indoor scene datasets [1, 2, 6, 9, 16] contain more information. For instance, ScanNet [9] supplies estimated camera parameters, surface segmentation, textured meshes and semantic segments; however, comparing with the proposed photogrammetry dataset Campus3D, these datasets generated by RGB-D sensors have their limitations of short measurement range and sensitivity to the sunlight’s infrared spectrum [14]. These natural limitations prevent the RGB-D datasets from applications of outdoor environment understanding.

Outdoor Dataset. Several outdoor scene 3D datasets [14, 25, 26, 31] are released in recent years. These datasets are generated via either MLS [3, 25, 26, 31] or TLS [14]. Points generated by the LiDAR are the raw output of the laser scanner, which are of high quality and large scales. The MLS point cloud datasets are always annotated with rich traffic elements to push the frontier of the autonomous driving field. One notable MLS point cloud dataset is a part of KITTI which was constructed by Geiger et al. [12, 13] and generated from 6 hours of traffic scenarios. Based on it, a point cloud dataset, semanticKITTI, has been proposed recently for outdoor semantic scene understanding [3]. However, different from our

1https://github.com/shinke-li/Campus3D
dataset collected by UAV imagery, LiDAR devices always suffer from occlusions thus lack for a holistic view of the scene.

**Deep Segmentation Model.** Semantic segmentation and instance segmentation are the major scene understanding tasks concerned by reconstruction. As the pioneering work PointNet and PointNet++ proposed by Qi et al. [23, 24], point-based deep neural models come to widely-studied in point cloud segmentation field since it can directly process the point cloud. Categories of point-based deep learning models mainly include feature pooling models [15, 23, 24, 43], convolution-based models [20, 30, 34], graph-based models [18, 33, 37] and attention-based models [41, 42]. Although most of these models are proposed for a single task, they can be involved in multitask learning. The examples are PointNet++ in ASIS [36] and PointNet in JSIS3D [22], which jointly learn instance embedding and semantic labeling in one structure. To jointly learn semantic labeling on multiple granularity levels, we propose our modification of point-based models fitting for multitask learning. The PointNet++ is applied as backbone since its general structure and high compatibility to multitask [35, 36].

### 3 CAMPUS3D DATASET

We note that the Campus3D is online accessible. Not only data can be downloaded there but also online interactive visualization and Github link for source codes are provided.

#### 3.1 Data Acquisition

The point cloud of Campus3D dataset was constructed by the technique of Structure from Motion with Multi-View Stereovision (SfM-MVS) [38] on UAV images. Here we briefly describe our workflow for obtaining the dataset. Firstly, we flew drones over all areas and took images with exact GPS coordinates. The device to capture imagery was DJI Phaton 4 Pro drones equipping cameras with a 1-inch 2 MP CMOS sensors, and the drone flight planning mobile apps used in our application were DJI GS Pro and Pix4D Capture. Then the points would be generated by photogrammetry processing and registration from captured images and coordinates using Pix4D as SFM-MVS software.

In image collection process, we applied two types of flight routing strategies for UAV photography: (1) grid and (2) circular, which were accessible in the drone flight planning mobile apps. For relatively high buildings, we applied multiple circular flights at different height levels. During UAV image capturing, the drone were flown when the clear view was guaranteed by the weather. More detailed settings can be found in the supplementary document.

#### 3.2 Data Annotation

To present more complicated geometric features, we annotated the point cloud with point-wise labels. In general, there are two approaches to perform 3D point-wise annotation: (1) label the pre-segmented clusters in 3D and (2) label the projected 2D image and assign labels to 3D points. Our strategy follows the second approach and performs a two-level of 2D projection segmentation, which avoids inherent error induced by pre-segmenting methods and lack of details in 2D projections of stationary angles. Initially, we divided the annotation tasks into hierarchical stages from coarse-grained to fine-grained label. In each stage, annotation was firstly done by 2D polygon partitions in three orthogonal view-angles. To refine the details, the obtained 3D partitions were then pruned in user-defined rotation angles. All the tasks were completed by opensource software CloudCompare [8] and its add-on functions. Multiple annotators were hired to perform above labeling task after taking training courses for days. To ensure the accuracy and consistency of annotations, we divided annotators to several groups, and work on labeling and verifying, respectively, for each stage. We require that every point is labeled at least three times by different annotators and verified to an exact label.

According to CityGML [17], objects from urban scene are modeled in different granularity levels defined by the LoD, which can cope with applications in different scales. Motivated by this concept, the category labels used in the Campus3D are constructed as a hierarchical structure with various granularity levels and displayed by Figure 3. The hierarchies of the structure can work similarly to the LoDs. Each label is formed based on two criteria: (1) geometric attribute and (2) geometrical attribute. They may mutually assist each other to parse the points into refined parts. For example of both “roof” and “driving_road” with plane structure, they are difficult to be distinguished in geometric features but need to be separated due to the semantic difference and practical function. All labels are self-explanatory except for the following ones. And we provide explanations for them: (1) “unclassified” refers to unrecognized or over-sparse regions. Instead of removing these data-points, this category is set for reserving the completeness of dataset; (2) “pathkstair”

#### Table 1: Comparison between Campus3D and popular scene-based point cloud datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data Source Type</th>
<th>Area/Length</th>
<th>Scene Type</th>
<th>Point #</th>
<th>Designed 3D Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScanNet [9]</td>
<td>RGB-D</td>
<td>floor: 34,453 m²</td>
<td>Indoor</td>
<td>-</td>
<td>Object classification; Instance &amp; semantic segmentation; CAD model retrieval</td>
</tr>
<tr>
<td>S3DIS[2]</td>
<td>RGB-D</td>
<td>6000 m²</td>
<td>Indoor</td>
<td>695.9M</td>
<td>Object detection</td>
</tr>
<tr>
<td>Matterport3D[6]</td>
<td>RGB-D</td>
<td>floor: 46,561 m²</td>
<td>Indoor</td>
<td>-</td>
<td>Instance &amp; semantic segmentation</td>
</tr>
<tr>
<td>Semantic3D[14]</td>
<td>Terrestrial Laser Scanner (TLS)</td>
<td>-</td>
<td>Outdoor</td>
<td>4,000M</td>
<td>Semantic segmentation</td>
</tr>
<tr>
<td>Paris-Lille-3D[25]</td>
<td>Velodyne HDL-32E (MLS)</td>
<td>1940 m</td>
<td>Outdoor</td>
<td>143.1M</td>
<td>Instance &amp; semantic segmentation</td>
</tr>
<tr>
<td>Campus3D (Ours)</td>
<td>UAV photogrammetry</td>
<td>1.58 ×10⁶ m²</td>
<td>Outdoor</td>
<td>937.1M</td>
<td>Hierarchical semantic segmentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Instance segmentation</td>
</tr>
</tbody>
</table>
is only for pedestrians while "driving_road" is only for vehicles; (3) "artificial_landscape" is referring to man-made landscape such as artificial pool while "others" represents some individual objects because there do not exist enough instances to group them as a new category. All the labels are defined in a rigid way for consistency of annotation.

3.3 Parsing and Statistics

To label the raw point clouds, we propose a hierarchical parsing method for decomposing the data into individually labeled points, which is naturally generated by the hierarchical annotation in previous section. The resulting Campus3D dataset can fulfill multiple tasks. We firstly divide the entire dataset into six identified regions: FASS, FOE, PGP, RA, UCC and YIH according to their architecture styles and functions. A descriptive summary of points of these six regions is given by Table 2.

Table 2: Area, mean height, points and points per area of data points of each region.

<table>
<thead>
<tr>
<th>Region</th>
<th>Area (m²)</th>
<th>Mean height (m)</th>
<th># of points</th>
<th># of points per area (m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FASS</td>
<td>276,746</td>
<td>48.74</td>
<td>114,599,515</td>
<td>414.10</td>
</tr>
<tr>
<td>FOE</td>
<td>247,924</td>
<td>49.76</td>
<td>34,347,821</td>
<td>138.54</td>
</tr>
<tr>
<td>PGP</td>
<td>277,468</td>
<td>50.19</td>
<td>29,595,347</td>
<td>106.66</td>
</tr>
<tr>
<td>RA</td>
<td>365,537</td>
<td>61.62</td>
<td>54,446,114</td>
<td>148.95</td>
</tr>
<tr>
<td>UCC</td>
<td>127,572</td>
<td>30.09</td>
<td>333,404,689</td>
<td>2613.46</td>
</tr>
<tr>
<td>YIH</td>
<td>284,775</td>
<td>42.08</td>
<td>354,491,876</td>
<td>1244.81</td>
</tr>
</tbody>
</table>

Due to the hierarchical annotation strategy, class labels of the Campus3D can be defined by a tree-like structure. Based on this structure, the coarse-grained level data can be simply obtained by merging their sub-class data including all leaf node, which is flexible for multi-level tasks. For example, class "building" data could be obtained by merging "wall" and "roof" data. After labeling each point by a hierarchical class tree, we performed instance labeling for each countable class, which may benefit 3D model reconstruction and scene understanding. For instance, to boost the LoD of the building model, it is necessary to distinguish various planar pieces from a roof. Figure 2 illustrates this parsing. We also note that more descriptive statistics of classes and instances are provided in the supplementary document.

3.4 Data Preprocessing

To practically perform the machine learning algorithms on the data, we need data simplification on point cloud with consideration of imbalanced density and processing efficiency. We provide a reduced dataset from the original points. This reduced dataset is voxel-sampled from the original dataset with a sampling size of 0.15 meter. The sampling method thins the data points and also inhibits the imbalanced distributions of points among different regions and instances, which is caused by the varies of morphology. Moreover, the 0.15m sample size can keep the smallest object in the whole campus. We term this dataset as Campus3D-reduced. Note that all experimental studies, scene understanding tasks and benchmarks in this paper are run on the Campus3D-reduced. Table 3 shows the training, validation and test splits. This splitting makes sure that training set and test/validation set have all types of instances. And the performance of the class "unclassified" is not included in current study, which follows the convention in this arena [20, 23, 24].
4 HIERARCHICAL LEARNING

In order to learn on hierarchical annotations of our dataset, we construct a five-level label tree displayed by Figure 3, where labels in each hierarchy can completely partition the entire dataset. In that case, each point possesses five parallel semantic labels, learning of which can be consider as a multi-label segmentation tasks. Compared with single label learning, the key problem towards hierarchical multi-label learning is how to leverage the relationship among hierarchies, while the hierarchical structure of labels should be kept. Therefore, we propose a simple yet effective framework, which includes a multi-task learning network and an ensemble process to maintain hierarchical structure. Before the methodology, we first proceed to the problem and performance metrics.

4.1 Problem and Metric Description

Let (C, ≤) represent the class hierarchy, where C is a set of classes and ≤ is a partial order representing the superclass relationship. For any c1, c2 ∈ C, c2 ≤ c1 if and only if c1 is a superclass of c2 or c1 = c2. Data point i with hierarchical annotation is denoted as (X, Y, S) with X ∈ X = R^D and S is a maximal chain of C. The problem of such label is that length of the label set |S| is not coherent from point to point. To construct a multi-label with coherent length, we further extend the definition of hierarchical learning by allowing duplication. We first note the set of all maximal elements in C by C_max and the set of all minimal elements C by C_min. Note that C_max and C_min both belong to AC, the set of all antichains in C. We define a relationship to compare the any two antichains named parent antichain:

**Definition 4.1 (Parent Antichain).** For two distinct sets Cc, Cp ∈ AC, if ∀c_i ∈ Cc, ∃c_i′ ∈ Cp let c_i ≤ c_i′, then Cp is called a parent antichain of Cc with notation Cc ≺ Cp.

Then we can obtain a sequence of antichains (sets) between C_min and C_max, namely, \((C_h)_{h=1}^H = (C_1 = C_{max}, C_2, \ldots, C_H = C_{min})\) with length \(H = \max\{|S| : \text{C}^H \text{ is coherent from point to point}\}\). On the sequence, a tree \(T\) can be constructed and displayed in Figure 3. The nodes in \(T\) can be associated with \(C_h\), while the edge is defined as the partial order relationship between classes. Now we define the hierarchical learning problem. A dataset \(D = \{(X_i, Y_i) : i \in \mathbb{Z}, 1 \leq i \leq N\}\), where N is the number of points, \(X_i \in X\) and \(Y_i \in Y = C^1 \times C^2 \times \ldots \times C^H\). The hierarchical learning problem is to learn a function \(f : X \rightarrow Y\) from a hierarchically annotated dataset \(D\).

Given a HL method \(f(\cdot)\), the performance can be evaluated by the conventional classification measurements such as accuracy, precision, recall, etc. However, they fail to take consistency into account which is critical for the HL. It is possible that a HL algorithm performs good in terms of conventional measurements, but generates highly inconsistent results violating the hierarchical relationship which are meaningless. Therefore, we propose a new measurement and quantitatively evaluate such consistency. Considering a solution (prediction) \(Y \in \mathcal{Y}\), we first define the fully consistent (FC) for a solution at Definition 4.1. The set of all FC solutions is denoted as \(\mathcal{Y}^{FC} (\mathcal{Y}^{FC} \subset \mathcal{Y})\), and it includes all paths from root to leaf nodes in tree \(T\). Based on it, we propose, consistency proportion (CP), to measure the consistency degree for solution \(Y^i\). The CP value is between 0 and 1, and being one represents a FC solution. Then, for a set of solutions \(\{Y^1, \ldots, Y^N\}\), the consistency rate (CR) is defined with parameter \(\alpha\) being the desired consistency level for each solution.

**Definition 4.2 (Fully Consistent).** Solution \(Y = (y_1, \ldots, y_H) \in \mathcal{Y}\) is defined as fully consistent (FC) if it satisfies \(y_H \leq y_{H-1} \leq \ldots \leq y_1\).

**Definition 4.3 (Consistency Proportion).** The consistency proportion (CP) of \(Y^i = (y^i_1, \ldots, y^i_H)\) is defined as:

\[
CP(Y^i) = \frac{\max_{\eta_1, \ldots, \eta_H} \sum_{h=1}^H \eta \left(\|y^i_h = y_h\|\right)}{H}, \tag{1}
\]

here \(I(x) = 1\) if \(x\) is True; \(0\) otherwise.

**Definition 4.4 (Consistency Rate).** The consistency rate (CR) with CP level \(\alpha\) for \(\{Y^1, \ldots, Y^N\}\) is:

\[
CR_{\alpha} = \frac{1}{N} \sum_{i=1}^N I(CP(Y^i) \geq \alpha). \tag{2}
\]

Here \(\alpha\) is a threshold parameter and \(0 \leq \alpha \leq 1\).

4.2 Methodology

We propose a two-stage framework to the HL (see Figure 4): (1) multi-task learning (MT) and (2) hierarchical ensemble (HE).

**Multi-task Learning (MT).** The main structure of MT networks contains a shared encoder and multiple parallel decoders with classification heads. To practically perform the MT, we utilized the feed forward architecture of PointNet++ [24]. Specifically, an feature map \(N \times D\) of point cloud with size \(N\) and feature dimension \(D\) is fed as input. Then the shared encoder encodes them as embedding. Such embedding is then decoded parallelly into \(N \times D^D\) by H decoders for H granularity levels. Decoder \(h\) computes the likelihood distributions of classes \((C_h)\) at granularity level \(h\) for each data point. The loss of MT method is the sum of the losses of its branches,

\[
L_{MT} = L_{\text{prediction}} + L_{\text{consistency}} \tag{3}
\]

where the prediction loss \(L_{\text{prediction}}\) is the weighted average of the cross entropy losses of H levels. And it is formulated as,

\[
L_{\text{prediction}} = \sum_{h=1}^H \beta_h \cdot L_{\text{prediction}}^h. \tag{4}
\]

Here, for granularity level \(h\), \(L_{\text{prediction}}^h\) and \(\beta_h\) are the cross-entropy loss and weight respectively. The consistency loss is served as a regularization term to maintain the consistence structure of predicted distributions

\[
L_{\text{consistency}} = \sum_{h=1}^H \gamma_h \sum_{y_h \in \text{PC}^h} \left[p_{h+1}(y_{h+1}) - p_{h}(y_{h})\right]^2. \tag{5}
\]

here \(\text{PC}^h = \{(y_h, y_{h+1}) : (y_1, \ldots, y_H) \in \mathcal{Y}^{FC}\}\), where \(\gamma_h\) is the loss weight of \(h\)-th level and \(p_{-1}(•)\) is the predicted likelihood distribution over class set (antichain) \(C_h\). By Definition 4.2, given FC solution \((y_1, \ldots, y_H)\), \(y_h\) is the superclass of or same as \(y_{h+1}\). This loss is
Hierarchical Ensemble (HE). The HE is a post-processing method for initial predicted results. It computes the weighted sum of likelihood scores over all the root-to-leaf paths in tree $\mathcal{T}$. The path associated with largest score is the final predicted solution. It is formulated as equation (7). Note that solutions generated by HE are FC and the CR (and/or CP) value is 1.

$$\hat{y}_{\text{HE}} = \arg \max_{(y_1, \ldots, y_H) \in \mathcal{Y}_{\text{RC}}} \sum_{k=1}^{H} p^k(y_k)$$  \hfill (7)

In order to perform comparison analysis, we also apply a multi-classifier (MC) method which does not leverage the mutual relationship across levels, and only trains an independent segmentation classifier for each granularity level. And $H$ classifiers are trained and evaluated separately. It performs conventional segmentation $H$ times for the dataset based on PointNet++. Furthermore, a variant of the proposed two-stage method, MC+HE is also investigated, which uses the HE to post-process outputs of the MC.

### 4.3 Experimental Results

Based on the class label tree given by Figure 3, we build five granularity levels ($H = 5$). They are given in the first and second columns of Table 5. PointNet++ [24] is used as backbone. We set $\beta_i = 1$ and $y_i = 0.05$ with $i \in \{1, \ldots, H\}$, and more detailed settings are present in the supplementary material. We apply CR1 ($\alpha = 1$), intersection-over-union (IoU) as well as overall accuracy (OA) for performance analysis.

#### Table 4: Test results (OA%) for different HL methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>$C^1$</th>
<th>$C^2$</th>
<th>$C^3$</th>
<th>$C^4$</th>
<th>$C^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC</td>
<td>85.3</td>
<td>79.5</td>
<td>78.3</td>
<td>76.3</td>
<td>74.0</td>
</tr>
<tr>
<td>MC+HE</td>
<td>89.1</td>
<td>81.4</td>
<td>79.9</td>
<td>77.9</td>
<td>73.5</td>
</tr>
<tr>
<td>MT</td>
<td>90.2</td>
<td>82.2</td>
<td>80.9</td>
<td>78.8</td>
<td>74.6</td>
</tr>
<tr>
<td>MT+HE</td>
<td>90.7</td>
<td>83.1</td>
<td>81.7</td>
<td>79.8</td>
<td>75.2</td>
</tr>
<tr>
<td>MT+HE+NC</td>
<td>90.2</td>
<td>83.1</td>
<td>81.7</td>
<td>80.0</td>
<td>75.4</td>
</tr>
</tbody>
</table>

Comparisons between different HL methods. After removing points with ground-truth label of “unclassified” (unlabeled), for each class, the intersection and union sets of predicted point set and ground-truth are generated; then the IoUs are computed as the ratio of intersection set cardinality to that of the union. And the OA is computed as proportion of correct predictions to total points. Results for test set results are presented in Table 4 and Table 5. There are several observations: (1) in terms of average IOU and OA of the granularity level, it decreases significantly with granularity level changing from $C^1$ to $C^5$. It indicates that the difficulty of the problem increases as the label instances become small and distributed sparsely; (2) the performance of MC + HE is better than that of MC only for most cases; (3) overall, the HL methods (i.e., MT + HE, MT+NC, MC + HE) taking hierarchical labels into account perform better than the MC without considering them. These observations demonstrate that hierarchical labels help and enhance the performance.

One possible reason of the better performance by the HL method is that the inherent relation among label layers provide additional geometrical information for semantic segmentation. A visual illustration is given by Figure 5. The MC semantic segmentation on the level $C^5$ and $C^3$ without other level information results in that “roof” is wrongly recognized as driving road (i.e.”road” or “not vehicle”) or natural ground “natural” (see $C^5$ (b) and $C^3$ (b) of Figure 5). We
found that they are geometrically similar but semantically different. Here we first define this phenomenon as geometric ambiguity: points with similar geometric features but significantly different semantic labels are wrongly classified to the same semantic class. As indicated by the result of MT ((c) column of Figure 5), hierarchical and multiple annotation can ameliorate this phenomenon. For the instance of C1 level in Figure 5, points on roof belonging to “construction” are easily recognized as “ground” by the MC, while the MT framework is able to segment them correctly by leveraging information from finer levels.

### Table 5: Test results (class IoU%) for HL methods

<table>
<thead>
<tr>
<th>Granularity Level</th>
<th>Class</th>
<th>Method</th>
<th>MC</th>
<th>MC+HE</th>
<th>MTnc</th>
<th>MT</th>
<th>MT+HE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>ground</td>
<td></td>
<td>78.9</td>
<td>83.5</td>
<td>84.9</td>
<td>85.5</td>
<td>85.5</td>
</tr>
<tr>
<td></td>
<td>construction</td>
<td></td>
<td>67.4</td>
<td>75.5</td>
<td>78.2</td>
<td>79.1</td>
<td>79.0</td>
</tr>
<tr>
<td>C2</td>
<td>natural</td>
<td></td>
<td>66.8</td>
<td>69.4</td>
<td>71.5</td>
<td>71.9</td>
<td>71.9</td>
</tr>
<tr>
<td></td>
<td>play_field</td>
<td></td>
<td>0.3</td>
<td>0.3</td>
<td>0.7</td>
<td>3.0</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>path/stair</td>
<td></td>
<td>7.2</td>
<td>8.2</td>
<td>7.8</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>driving_road</td>
<td></td>
<td>49.9</td>
<td>52.0</td>
<td>51.2</td>
<td>52.5</td>
<td>52.3</td>
</tr>
<tr>
<td></td>
<td>construction</td>
<td></td>
<td>72.9</td>
<td>75.5</td>
<td>77.0</td>
<td>79.1</td>
<td>79.0</td>
</tr>
<tr>
<td>C3</td>
<td>natural</td>
<td></td>
<td>67.1</td>
<td>69.4</td>
<td>71.8</td>
<td>71.9</td>
<td>71.9</td>
</tr>
<tr>
<td></td>
<td>play_field</td>
<td></td>
<td>0.8</td>
<td>0.3</td>
<td>0.5</td>
<td>1.9</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>path/stair</td>
<td></td>
<td>7.8</td>
<td>8.2</td>
<td>7.9</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>vehicle</td>
<td></td>
<td>34.5</td>
<td>38.7</td>
<td>36.7</td>
<td>36.4</td>
<td>36.6</td>
</tr>
<tr>
<td></td>
<td>not vehicle</td>
<td></td>
<td>48.6</td>
<td>51.1</td>
<td>50.1</td>
<td>51.1</td>
<td>51.3</td>
</tr>
<tr>
<td></td>
<td>building</td>
<td></td>
<td>70.1</td>
<td>72.1</td>
<td>74.0</td>
<td>75.7</td>
<td>76.0</td>
</tr>
<tr>
<td></td>
<td>link</td>
<td></td>
<td>2.1</td>
<td>2.2</td>
<td>3.7</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>facility</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>C4</td>
<td>natural</td>
<td></td>
<td>71.0</td>
<td>69.4</td>
<td>72.2</td>
<td>71.7</td>
<td>71.9</td>
</tr>
<tr>
<td></td>
<td>play_field</td>
<td></td>
<td>1.6</td>
<td>0.3</td>
<td>2.0</td>
<td>1.8</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>sheltered</td>
<td></td>
<td>10.7</td>
<td>10.4</td>
<td>10.7</td>
<td>1.4</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>unsheltered</td>
<td></td>
<td>4.4</td>
<td>4.4</td>
<td>4.7</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>bus_stop</td>
<td></td>
<td>0.4</td>
<td>0.5</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>car</td>
<td></td>
<td>40.7</td>
<td>39.9</td>
<td>38.5</td>
<td>37.0</td>
<td>36.6</td>
</tr>
<tr>
<td></td>
<td>not vehicle</td>
<td></td>
<td>50.6</td>
<td>51.1</td>
<td>50.1</td>
<td>51.0</td>
<td>51.3</td>
</tr>
<tr>
<td></td>
<td>wall</td>
<td></td>
<td>57.3</td>
<td>56.0</td>
<td>57.2</td>
<td>57.1</td>
<td>57.5</td>
</tr>
<tr>
<td></td>
<td>roof</td>
<td></td>
<td>58.6</td>
<td>57.2</td>
<td>60.4</td>
<td>61.1</td>
<td>61.3</td>
</tr>
<tr>
<td></td>
<td>link</td>
<td></td>
<td>3.5</td>
<td>2.2</td>
<td>3.6</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>artificial_landscape</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>lamp</td>
<td></td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>others</td>
<td></td>
<td>0.2</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Insights of Consistency Rate.** As a metric evaluating consistency of hierarchical relationship, the result of CR1 reveals that our framework leverages the mutual assistance among hierarchies. From Figure 6, the CR1 of MT is over around 15% than that of MC which ignores the hierarchical annotation. It suggests that MT learning may correct the results in certain level according to features from other granularity levels. On another hand, with comparison of MC and MC+HE results from Table 5 and Table 4, HE also boosts the performance by maintaining hierarchical relationship forcibly, while this boosting is not significant from MT to MT+HE. The results of CR1 quantitatively explain why HL methods can effectively address the geometric ambiguity discussed above.

**Effectiveness of Consistency Loss.** Performance differences between MT and MTnc in Figure 6 and Table 4 demonstrate the effectiveness of the proposed consistency loss. With it, MT can significantly restrain the hierarchical violations in segmentation results, while MTnc, ignoring it, results in around 10% decrease in terms of CR1. Moreover, MT also performs better than MTnc in terms of OA (see Table 4).

5 HIERARCHICAL SCENE UNDERSTANDING TASKS AND BENCHMARKS

In this section, we apply the HL framework for scene understanding and build benchmarks on two tasks of hierarchical semantic segmentation and instance segmentation. We first investigate the effectiveness of sampling methods for feature learning on large-scale point cloud datasets.

### 5.1 Sampling Methods

For scene based point cloud datasets, sampling is necessary for feature learning because of requirements of efficiency and fixed size of model input. Typical sampling strategies are uniform sampling and farthest point sampling (FPS) [23]. Here we do not apply farthest point sampling (FPS) due to its computational inefficiency. The simplest sampling method is to randomly pick a fixed size of points with uniform distribution. However, randomly sampling a small size of points from large point cloud data would induce...
considerable randomness into the samples, which may lead to failure of training process. Therefore, to conduct less bias and learnable sampling, we experimented two variations of uniform sampling, the details of which are presented in the supplementary document: (1) $l$-w random block sampling ($l$-w RBS), and (2) random centered K nearest neighbor (RC-KNN). Given a set of points $\mathcal{D}_x = \{(x_1^1,x_2^1,x_3^1),(x_1^2,x_2^2,x_3^2),\ldots,(x_1^n,x_2^n,x_3^n)\} \subseteq \mathbb{R}^3 (D = 3$ and three coordinates: latitude, longitude and height), we define them and illustrate how to select $N (N < n)$ points from $\mathcal{D}_x$ as follows.

$l$-w RBS randomly chooses a point $P_c (x_1^c,x_2^c,x_3^c)$ from $\mathcal{D}_x$ with an uniform distribution and then uniformly samples $\mathcal{D}_x' = \{(x_1^1,x_2^1,x_3^1),(x_1^2,x_2^2,x_3^2),\ldots,(x_1^i,x_2^i,x_3^i),(x_1^{i+1},x_2^{i+1},x_3^{i+1})\}$ in a $l$-w block centered at $P_c$, namely, for any $1 \leq j \leq N-1$, $x_1^j - \frac{l}{2} \leq x_1^j \leq x_1^j + \frac{l}{2} \leq x_1^j$ and $x_2^j - \frac{w}{2} \leq x_2^j \leq x_2^j + \frac{w}{2}$.

RC-KNN randomly chooses a point $P_c (x_1^c,x_2^c,x_3^c)$ from $\mathcal{D}_x$ with an uniform distribution, and then $K = \frac{N}{k}$ nearest neighbors to point $P_c$ in terms of Euclidean distance are chosen as the sampled points.

In order to investigate the effectiveness of the above two sampling methods on the HL methods, we apply them with PointNet++ [24] as the feature leaning network on our dataset. Specifically, in each training iteration, we use either RBS or RC-KNN to select $N$ points from a randomly picked region in training set as a sample in a batch. The sample size $N$ is set as 2048, and block size of both $l$ and $w$ in RBS are set as 12m. We compute the mean IoU (mIoU) across all classes for each granularity level. The test results are given in Table 6. From the results, we can see that the RBS method dominates the RC-KNN method by our setting. Note that the setting of RBS sampling is also utilized in Section 4.

Table 6: Semantic segmentation results (mIoU%) for RBS and RC-KNN sampling methods.

<table>
<thead>
<tr>
<th>Model Method</th>
<th>Sampling Method</th>
<th>$C^1$</th>
<th>$C^2$</th>
<th>$C^3$</th>
<th>$C^4$</th>
<th>$C^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT</td>
<td>RBS</td>
<td>82.3</td>
<td>68.6</td>
<td>41.3</td>
<td>29.7</td>
<td>20.1</td>
</tr>
<tr>
<td></td>
<td>RC-KNN</td>
<td>77.4</td>
<td>61.7</td>
<td>37.2</td>
<td>25.5</td>
<td>9.9</td>
</tr>
<tr>
<td>MT + HE</td>
<td>RBS</td>
<td>82.2</td>
<td>68.5</td>
<td>41.1</td>
<td>29.8</td>
<td>20.1</td>
</tr>
<tr>
<td></td>
<td>RC-KNN</td>
<td>77.0</td>
<td>62.0</td>
<td>37.1</td>
<td>25.6</td>
<td>9.8</td>
</tr>
</tbody>
</table>

5.2 Semantic Segmentation

In this section, we benchmark the performance on semantic segmentation task. Three established models are applied: PointNet++ [24], PointCNN [20] and DGCNN [37]. The sampling method used here is the RBS sampling with block size of 12m ($l = w = 12m$). We take the hierarchical annotation into account and apply our proposed MT+HE method. The mIoU across all classes for each granularity is used as the performance metric. The results of both test and validation are given by Table 7.

Table 7: Semantic segmentation results (mIoU%) for three feature learning models with HL methods.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Learning Model</th>
<th>$C^1$</th>
<th>$C^2$</th>
<th>$C^3$</th>
<th>$C^4$</th>
<th>$C^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>PointNet++</td>
<td>79.7</td>
<td>67.0</td>
<td>43.4</td>
<td>33.4</td>
<td>21.9</td>
</tr>
<tr>
<td></td>
<td>PointCNN</td>
<td>86.9</td>
<td>77.4</td>
<td>51.1</td>
<td>38.0</td>
<td>27.2</td>
</tr>
<tr>
<td></td>
<td>DGCNN</td>
<td>88.1</td>
<td>79.8</td>
<td>53.0</td>
<td>39.5</td>
<td>29.5</td>
</tr>
<tr>
<td>Test</td>
<td>PointNet++</td>
<td>79.7</td>
<td>67.0</td>
<td>43.4</td>
<td>33.4</td>
<td>21.9</td>
</tr>
<tr>
<td></td>
<td>PointCNN</td>
<td>88.6</td>
<td>78.2</td>
<td>58.4</td>
<td>41.1</td>
<td>27.2</td>
</tr>
<tr>
<td></td>
<td>DGCNN</td>
<td>90.9</td>
<td>81.3</td>
<td>61.5</td>
<td>46.3</td>
<td>29.1</td>
</tr>
</tbody>
</table>

5.3 Instance Segmentation

In this section, we build the instance segmentation benchmark of current dataset. The training, validation and test splitting still follows Table 3. We perform this task for the granularity level four ($C^4$) only, where there exists the largest number of available classes and instances among all granularity levels for training, validation and test. The ASIS [36] and SGPN [35] method were used here for the baseline evaluation. For each class, the weight coverage (WCov) as introduced by Wang et al. in [36] is computed as the performance measurement. Results for both validation and test sets are shown in Table 8, which shows that ASIS [36] performs better than SGPN [35]. Note that classes “natural”, “path&stairs”, “not vehicle” and “facility” are not countable, thus no instance segmentation results are for them.

Table 8: Instance segmentation results (WCov) for selected classes at granularity level four ($C^4$).

<table>
<thead>
<tr>
<th>Instance class name</th>
<th>Validation ASIS</th>
<th>Validation SGPN</th>
<th>Test ASIS</th>
<th>Test SGPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>play_field</td>
<td>0.0</td>
<td>0.0</td>
<td>3.4</td>
<td>0.0</td>
</tr>
<tr>
<td>vehicle</td>
<td>34.0</td>
<td>17.3</td>
<td>44.1</td>
<td>32.7</td>
</tr>
<tr>
<td>building</td>
<td>53.4</td>
<td>35.7</td>
<td>40.3</td>
<td>32.1</td>
</tr>
<tr>
<td>link</td>
<td>13.3</td>
<td>10.1</td>
<td>11.2</td>
<td>8.5</td>
</tr>
<tr>
<td>Average</td>
<td>26.4</td>
<td>15.8</td>
<td>24.8</td>
<td>18.3</td>
</tr>
</tbody>
</table>

6 CONCLUSION

A well-annotated point cloud dataset with two benchmarks, Campus3D, is proposed in this paper. It is annotated with multiple and hierarchical label for the better scene understanding and potential usage in reconstruction. We define the HL problem and propose a new measure to evaluate the consistency across granularity levels. A two-stage method MT+HE is presented to the HL. Experimental results demonstrate its effectiveness comparing with MC without taking multiple and hierarchical information into account. Moreover, we investigate two sampling methods for point cloud learning with HL methods and identify RBS as the useful one. Future users will benefit from these initial and basic explorations. In the end, we apply established models and benchmark performance for semantic and instance segmentation for future comparisons. Other potential tasks can be built based on the Campus3D such as hierarchical instance segmentation and 3D model reconstruction.

ACKNOWLEDGMENTS

This research is supported by National University of Singapore (NUS) Institute of Operations Research and Analytics (IORA) grant R-726-000-002-646 and National Research Foundation of Singapore grant NRF-RSS2016-004. The authors gratefully acknowledge the data collection support of Virtual NUS team.
REFERENCES


Hengshuang Zhao, Li Jiang, Chi-Fung Fu, and Jiaya Jia. 2019. PointWeb: Enhancing local neighborhood features for point cloud processing.