Computational Visual Media
https://doi.org/10.1007/s41095-022-0311-7 Vol. 10, No. 3, June 2024, 425–438

Research Article

Active self-training for weakly supervised 3D scene semantic segmentation

Gengxin Liu[1], Oliver van Kaick[2], Hui Huang[1], and Ruizhen Hu[1] (✉)

Abstract Since the preparation of labeled data for training semantic segmentation networks of point clouds is a time-consuming process, weakly supervised approaches have been introduced to learn from only a small fraction of data. These methods are typically based on learning with contrastive losses while automatically deriving per-point pseudo-labels from a sparse set of user-annotated labels. In this paper, our key observation is that the selection of which samples to annotate is as important as how these samples are used for training. Thus, we introduce a method for weakly supervised segmentation of 3D scenes that combines self-training with active learning. Active learning selects points for annotation that are likely to result in improvements to the trained model, while self-training makes efficient use of the user-provided labels for learning the model. We demonstrate that our approach leads to an effective method that provides improvements in scene segmentation over previous work and baselines, while requiring only a few user annotations.

Keywords semantic segmentation; weakly supervised; self-training; active learning

1 Introduction

Recent years have seen the introduction of approaches for semantic segmentation of point clouds, which have been quite successful in providing meaningful segmentations of indoor scenes [1–4]. Much of this success is due to the use of deep learning methods combined with the availability of large amounts of labeled data, e.g., datasets such as ScanNet [5] and S3DIS [6]. However, the applicability and scalability of these methods to new contexts is limited, since creating training data is a time-consuming task, involving the manual labeling of points.

To address the dependency of segmentation methods on large amounts of training data, methods for weakly supervised segmentation have been introduced, which require only a fraction of the training data commonly used. These methods either estimate pseudo-labels for the data in order to train segmentation networks [7–9], or use variations of a contrastive loss for enabling learning transfer [10–12].

The recent "one thing one click" method [14] introduces an iterative self-training approach that alternates between training and label propagation, where the labels from points annotated by the user are propagated to unlabeled data based on a learned data similarity measure. The method achieves some of the best results among weakly supervised methods by training on data with only one label per object. However, the user is responsible for manually selecting one point per object, which can be difficult in cluttered scenes. Moreover, the method is quite complex, involving the combination of two networks, one for semantic segmentation and the other for similarity learning, trained in an iterative manner with a contrastive loss. The extra complexity of training a similarity estimation network is necessary for accurate label propagation during self-training.

In this paper, we introduce a method for weakly supervised segmentation that combines self-training with active learning [15–17]. Our focus is on improving the selection of samples to be annotated while simplifying the label propagation step. The

1 College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, China. E-mail: G. Liu, 2100271019@email.szu.edu.cn; H. Huang, huihuang@szu.edu.cn; R. Hu, ruizhen.hu@szu.edu.cn (✉).

2 School of Computer Science, Carleton University, Ottawa K1S 5B6, Canada. E-mail: Oliver.vanKaick@carleton.ca.

Manuscript received: 2022-06-16; accepted: 2022-09-04

active learning method we introduce automatically selects the points that have to be annotated by the user according to an uncertainty measure, reducing the amount of work involved in the annotation task, and querying the user for annotations of the points likely to lead to considerable improvements in the model's results. In addition, our self-training method is much simpler, requiring only the training of a segmentation network. We perform label propagation based on scene geometry via super-voxels, without the need to train a similarity estimation network, which is unnecessary for the final segmentation task.

We demonstrate that active learning combined with our simpler self-training pipeline leads to improved point cloud segmentation results for indoor scenes, compared to previous approaches and baselines. As Fig. 1 shows, we obtain higher mIoUs than previous works on two established datasets of indoor scenes: ScanNet and S3DIS. In addition, we show that selecting points according to active learning, which can potentially query multiple points for the most challenging objects to segment, leads to improved results compared to selecting a single point per object, while still requiring the same number or fewer user annotations.

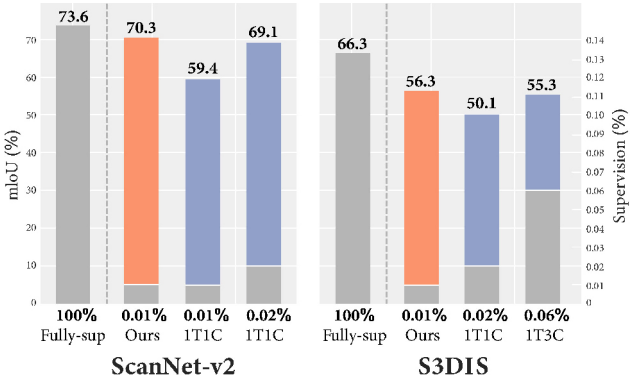


Fig. 1 Comparison of our weakly supervised semantic segmentation method to Liu et al. (denoted 1T1C or 1T3C depending on the amount of annotation data used) and a fully-supervised method with the same backbone as ours [13], on two datasets of 3D scenes. Our method achieves better results (left axis) while using an equal or smaller number of user annotations than other weakly supervised methods (right axis).

2 Related work

2.1 Understanding 3D scenes

3D scenes are commonly scanned from the environment and represented as point clouds, and their understanding involves solving problems such as 3D object detection, classification, semantic segmentation, and instance segmentation. The problem most related to our paper is semantic segmentation of point clouds, and thus we discuss it in more detail here.

Earlier solutions for scene segmentation transform point clouds into volumetric grid representations for processing with convolutional neural networks (CNNs) [18]. However, since it is unnatural to represent sets of points as a volumetric grid, several approaches have been introduced for directly processing point clouds, such as variations of PointNet [1, 19] that make use of the symmetric max pooling operation, or generalizations of convolution to sets of points, such as Li et al., Thomas et al., Wu et al. [20], Komarichev et al. [21], Su et al. [22], and hybrid approaches such as Liu et al. [23] and Han et al. [4]. Ye et al. [24] use two-direction hierarchical recurrent neural networks to extract long-range spatial dependencies in the point cloud. Guo et al. [25] design a point cloud processing architecture comprised of transformers. Peng et al. [26] propose a part-level semantic segmentation annotation method for a single-view point cloud using the guidance of labeled synthetic models.

A few methods are also based on other data representations, such as multiple 2D views [27], combined 2D/3D information [28], hash tables that enable sparse convolution [29], and graphs [30]. Tatarchenko et al. [31] project features into predefined regular domains and apply 2D CNNs to the domain. Some works focus on online segmentation, which aims to perform real-time 3D scene reconstruction along with semantic segmentation [32, 33]. We base the backbone of our segmentation network on the 3D U-Net architecture [13], which implements the efficient *generalized sparse convolution*.

2.2 Weakly supervised segmentation

In our paper, we address the weakly-supervised semantic segmentation of point clouds with limited annotations. Recently, a few methods have been introduced that have made significant advances in solving this problem. Xu et al. directly label a small number of points (around 10% of the data) and train an incomplete supervision network with spatial smoothness constraints, showing that the learning gradient of the insufficient supervision approximates the gradient of the full supervision. Other methods

generate pseudo-labels for the unlabeled data based on a small set of labeled samples. For example, Wei et al. perform weak supervision with labels that only indicates which classes appear in the training samples. These labels are transformed into point pseudo-labels with a region localization method. Cheng et al. [34] propagate sparse point labels to the unlabeled data using a graph of superpoints extracted from the point cloud.

Another important line of work investigates the use of *contrastive losses* for learning transfer when processing point clouds. Xie et al. perform the first studies in this regard, showing that learning transfer is possible for point cloud processing. Moreover, Jiang et al. use a contrastive loss to guide semantic segmentation, while Hou et al. extend the contrastive loss to integrate spatial information. Zhang et al. [35] use a self-supervised method in a contrastive framework to pre-train point cloud processing networks with single-view depth scans. The networks can then be fine-tuned for multiple tasks such as segmentation and object detection. Liu et al. combine self-training based on contrastive learning with a label-propagation mechanism in the "one thing one click" method, achieving some of the best results for point cloud segmentation. In contrast, we introduce a method that provides more effective selection of labeled samples by active learning, while simplifying label propagation, achieving significant improvements over previous works.

2.3 Active learning

Active learning approaches seek to minimize manual annotation effort by strategically querying the user for those annotations that maximize the improvement of the learned models. This is typically an iterative process involving the selection of points to be labeled and then updating a model based on the new annotations. Earlier active learning approaches for segmentation focused on the annotation of 3D shapes, such as the method of Yi et al. [36], querying users for annotation and verification. More recent approaches focus on the annotation of point clouds of scanned objects, e.g., as in the method of Hu et al., simultaneously performing reconstruction and segmentation.

For point clouds of entire scenes, Wu et al. introduce an active learning approach that measures the uncertainty in the point cloud labeling and selects diverse points to minimize redundancy in the point selection. Shi et al. maximize model performance for a limited annotation budget by measuring consistency at the super-point level. We also incorporate iterative active learning into our method, although we measure uncertainty based on the stochastic behavior of a segmentation network, leading to an effective sample selection for segmentation.

3 Self-training with active learning

3.1 Overview

An overview of our method is illustrated in Fig. 2. Our goal is to train a 3D semantic segmentation network (3D U-Net in Fig. 2) that is able to predict accurate semantic labels for the input point cloud X representing a 3D scene. Our key insight is

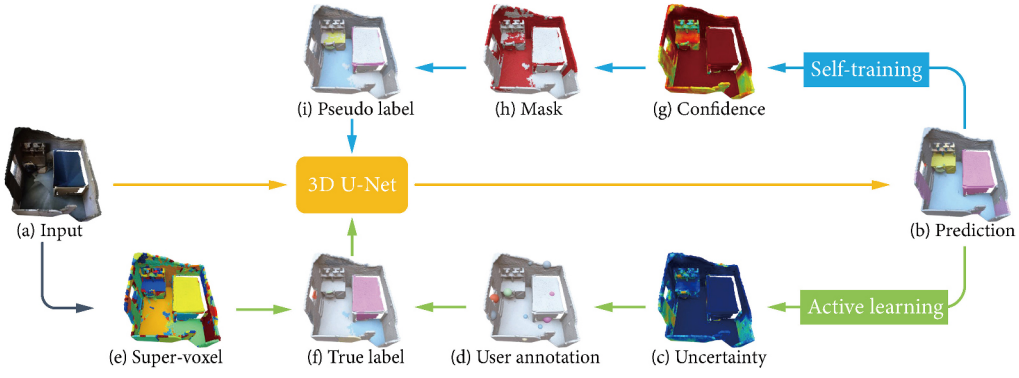


Fig. 2 Overview of our method: (a) Given an input scene, we train a neural network to predict (b) a semantic segmentation of the point cloud. Based on the uncertainty of the prediction (c), an active learning method selects a small set of samples which are (d) annotated by a user, where the selected samples are shown as big dots and the samples of previous iterations are shown as small dots. The labels are (f) propagated to the entire point cloud based on (e) an over-segmentation of the points into super-voxels. At the same time, a self-training method selects (i) a set of pseudo labels with (h) the mask determined based on (g) label confidence. The network is then refined based on the (f) true propagated and (i) pseudo labels. This process is then repeated for multiple iterations until a budget of annotations is achieved.

that, when training the segmentation network with extremely sparse labels, the selected annotation labels have a substantial impact on the accuracy of the segmentation model. Thus, the segmentation network is trained in an iterative manner with self-training combined with active learning, according to the following steps.

As a pre-processing step, given the input point cloud X, as shown in Fig. 2(a), we first over-segment X into geometrically homogeneous super-voxels. The super-voxels are used for label propagation: all points in the same super-voxel share the label provided by the user for a point in the super-voxel. Note that active learning ensures that the user is never required to annotate more than one point per super-voxel.

During each iteration of self-training, the segmentation network is trained with two sets of per-point labels: T and P, as in Figs. 2(f) and 2(i). The set of true labels T is obtained from user annotations of a sparse set of sample points $\hat{T}$. Based on the uncertainty predicted by the network trained in the previous iteration, we sample a set of points for user annotation.

The true labels $\hat{T}$ of these samples are then propagated within their containing super-voxels to yield a set of propagated per-point labels T, as shown in Figs. 2(d)–2(f). The set of pseudo labels P is taken from the prediction results of the segmentation network trained in the previous iteration of the self-training, where labels are selected based on the prediction confidence, and is empty in the first iteration.

We repeat the iterations composed of user annotation and network training until reaching a predefined number of iterations, to satisfy a requested annotation budget. We provide more details of the components of the method in the following.

3.2 3D semantic segmentation network

We adopt the 3D U-Net architecture of Choy et al. as the backbone of our segmentation network. The input to U-Net is a point cloud X of N points, with each point x_i containing both 3D coordinates and color information, where $i \in \{1, \ldots, N\}$. The network predicts the probability of each semantic category for each point x_i, denoted as $p_{i,c}$; the probability corresponding to the ground truth category $\bar{c}$, provided either by T or P, is denoted as $p_{i,\bar{c}}$. The network is then trained with the softmax cross-entropy loss:

$$L = \frac{1}{|T|} \sum_{i \in T} -\log p_{i,\bar{c}} + \lambda \frac{1}{|P|} \sum_{i \in P} -\log p_{i,\bar{c}} \quad (1)$$

where λ is a combination weight.

In the first iteration of self-training, the network is trained with the set T derived from a set of randomly sampled points $\hat{T}$ annotated by the user, and the set P is empty. In subsequent iterations, the set $\hat{T}$ is expanded with new samples annotated by users, where the samples are selected via active learning as explained in Section 3.3, which are then propagated through the super-voxels to form a new set T. The set P is updated with the prediction results of the current network, as explained in Section 3.4.

Note that, during inferencing, the average of predictions of all points inside the same super-voxel is used as the prediction result for the super-voxel, so that all the points in the same super-voxel have the same prediction. This "voting method" ensures that the prediction for the points inside each super-voxel is consistent, which improves the final prediction accuracy, as we show in our ablation studies.

3.3 Active learning for true label annotation

During the active learning process, the user is asked to annotate a sparse set of points with labels, as illustrated in Fig. 2(d). To select the most effective set of points to annotate for improving the accuracy of the segmentation, we measure the uncertainty of the labeling of each point based on current prediction results. The uncertainty of each point is measured by calculating the standard deviation of several stochastic forward passes, and using the one corresponding to the category with the highest mean prediction confidence. More specifically, for each input point cloud, we first get K different versions via the standard data augmentation operations of Choy et al., and then for each point, we compute the mean and standard deviation of these K probability distributions predicted from those K different input versions. Finally, the standard deviation of the category with the highest mean probability is used as the uncertainty u_i of the point x_i:

$$u_i = \sqrt{\frac{\sum_k (p^k_{i,\hat{c}} - \overline{p}_{i,\hat{c}})^2}{K}} \quad (2)$$

where $p^k_{i,c}$ is the predicted probability of point x_i in the k-th point cloud version for category c and $\overline{p}_{i,c} = \left(\sum_k p_{i,c}\right) \Big/ K$ is the mean probability for the

K versions for category c, with $\hat{c}$ being the category with the highest mean probability.

For each iteration, we select m points according to the uncertainty distribution over points, since intuitively points with high uncertainty require more reliable user input.

3.4 Pseudo label generation

Following Liu et al., we iteratively update the set of pseudo labels P. Starting with the label predictions of the segmentation network trained at a given iteration, we take the predictions with high confidence (larger than a given threshold τ) and use them as updated pseudo labels P. The pseudo labels P are then used together with the labels T, propagated from the true labels $\hat{T}$, to train the network in the next iteration. Note that we also use the mean prediction probability $\overline{p}_{i,\hat{c}}$ of K different point cloud versions, as in Eq. (2), to compute the confidence in this step.

To limit error propagation in our iterative training and pseudo-labeling process, we generate new labels for all unlabeled samples and reinitialize the neural network after each pseudo-labeling step, following Rizve et al. [37].

4 Results and evaluation

4.1 Datasets

Our experiments were conducted on ScanNet-v2 and S3DIS, which allowed us to compare our results directly to those of Liu et al. and other methods. We used the original training–validate–test split provided in these two datasets. One thing to note is that we focus on the "data efficient" annotation setting as in Hou et al., which is a more realistic setting than the "one thing one click" setting in Liu et al., since Liu et al. require the user to identify each individual object in the scene. Regarding super-voxel creation, for the ScanNet dataset, we used the method of Dai et al., while for S3DIS, we used the method of Landrieu et al., following Liu et al.

4.2 Implementation details

We implemented our method with the PyTorch [38] framework based on the implementation of Choy et al. We used the default data augmentation of Choy et al. and set the batch size to 4 for both ScanNet-v2 and S3DIS datasets. The number of training iterations on Scannet-v2 and S3DIS were 6 and 5, respectively. For ScanNet-v2, the initial learning rate was 0.1, with polynomial decay with power 0.9, and the model was trained for a total of 100k steps in each iteration. For S3DIS, the initial learning rate was 0.03, with polynomial decay with power 0.9, and the model was trained for a total of 60k steps in each iteration. Uncertainty and confidence were computed from $K = 5$ different versions' predictions. The threshold τ for pseudo label generation was set to 0.99 in the first few iterations and 0.95 in the last two iterations, to generate more training data with the refined segmentation network. The combination weight λ was set to 0.5.

In the following sections, we first compare results with existing methods in Section 4.3 and give results of ablation studies in Section 4.4, for both datasets. Then, in Section 4.5, we show that our method can also work in the "one thing one click" setting and obtain better results.

4.3 Comparison to existing methods

4.3.1 Notes

For a fair comparison, we used fewer or the same number of user annotations as existing methods. The actual number of user annotations used is reported in each table. As our method uses active learning to select samples to annotate, the number of samples is evenly distributed, i.e., if n is the total number of user annotations and k is the number of iterations, then $m = n/k$ points are sampled in each iteration for users to annotate.

4.3.2 Results on ScanNet-v2

Table 1 reports results on the ScanNet-v2 test set, where the existing methods are roughly divided

Table 1 Comparison of our method ActiveST to existing methods and to our fully-supervised baseline on the ScanNet-v2 test set

Method	Supervision	mIoU (%)
Qi et al. [1]	100%	33.9
Tatarchenko et al.	100%	43.8
Thomas et al.	100%	68.4
Graham et al.	100%	72.5
Choy et al.	100%	73.6
Kundu et al.	100%+2D	74.6
Our fully-sup baseline	100%	73.6
Liu et al.	0.02%	69.1
Hou et al.	20 points/scene	53.1
Xie et al.	20 points/scene	55.0
Liu et al.	20 points/scene	59.4
ActiveST (ours)	20 points/scene	**70.3**

into two groups: (i) fully supervised approaches with 100% supervision, including the state-of-the-art networks for point cloud segmentation, and (ii) weakly supervised approaches, including the most recent work of Liu et al. and methods using contrastive pre-training followed by fine-tuning with limited labels. Note that "our fully-supervised baseline" refers to the segmentation network of Choy et al. which we take as the backbone of our method, trained with 100% supervision.

Our method produces competitive results with only 20 labeled points per scene. Firstly, our method outperforms the best weakly supervised approaches in the "data efficient" setting with 20 points by nearly 11% mIoU. Our method also surpasses that of Liu et al. by about 1% mIoU whereas Liu et al. use twice the number of user annotations, and requires object instance information. Secondly, the gap between our method and full supervision is less than 3.3% mIoU, showing the effectiveness of our method.

Figure 3 gives visual examples of results obtained with our method and the fully-supervised baseline, compared to the ground truth. We can see that our method obtains results comparable to, and sometimes even better than, the fully-supervised baseline. For example, for the scene shown in the first row, our

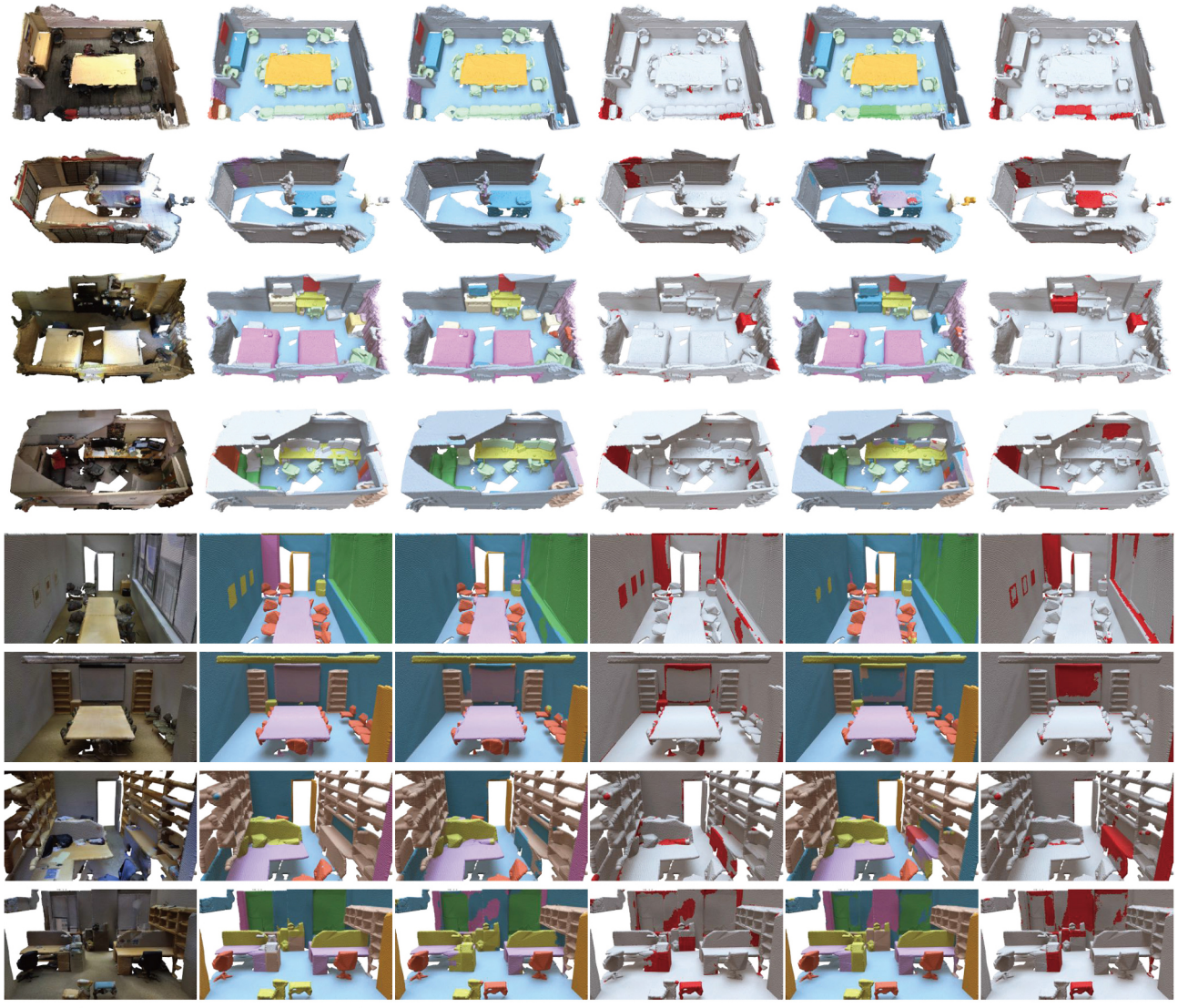


Fig. 3 Qualitative segmentation results: the first four rows are from the ScanNet-v2 dataset and the following four rows are from the S3DIS dataset. The input and the ground truth segmentations are presented in (a) and (b). (c) and (e) are the prediction of our method and of the fully-supervised baseline, while (d) and (f) are the corresponding error maps, where red regions indicate incorrect predictions.

method correctly labels all the chairs in different arrangements. Our method is also able to recognize the cabinet for the scene in the second row, although the fully-supervised baseline misclassifies it as a counter.

4.3.3 Results on S3DIS

Table 2 reports the comparison to existing methods on the S3DIS dataset, for both fully supervised approaches and weakly supervised approaches. For the latter, we only compared to the method of Liu et al., as it is the most recent work and works better than previous approaches.

We can see that with only 20 labeled points per scene (0.01% supervision), our method outperforms the method of Liu et al. with 0.02% supervision by nearly 6% mIoU. It also surpasses by nearly 1% mIoU the method of Liu et al. with 0.06% supervision that annotates 3 random points per object. Furthermore, our approach even outperforms several fully-supervised methods, again demonstrating the effectiveness of our method.

Figure 3 shows visual results of our method on the S3DIS dataset. We see that even with more challenging and crowded scenes, our method can still obtain reasonably good prediction results similar to those obtained from the fully-supervised baseline.

4.4 Ablation studies

4.4.1 Results

Our ablation studies were conducted using the most challenging setting with only 20 points annotated in each scene, for both the ScanNet-v2 and S3DIS datasets. For ScanNet-v2, evaluation was conducted on the validation set. For S3DIS, evaluation was conducted on Area 5.

Table 2 Comparison of our method ActiveST to existing methods and to our fully-supervised baseline on Area 5 of S3DIS

Method	Supervision	mIoU (%)
Qi et al. [1]	100%	41.1
Tatarchenko et al.	100%	52.8
Ye et al.	100%	53.4
Landrieu et al.	100%	58.0
Choy et al.	100%	66.3
Kundu et al.	100%+2D	65.4
Our fully-sup baseline	100%	66.3
Liu et al.	0.06%	55.3
Liu et al.	0.02%	50.1
ActiveST (ours)	20 points/scene	**56.3**

We tested the two key components of our method: self-training and active learning. We also tested the voting process used during the inference. The results are shown in Table 3. We see that each component of our method contributes to the quality of the final results.

More specifically, in the first row, we show the results of our baseline method, which is the segmentation network of Choy et al. trained with all user annotations, where the samples are either selected with a pre-trained model provided by Hou et al. for ScanNet-v2 or randomly selected for S3DIS. Adding voting for predictions within each super-voxel improves the results, confirming the idea of maintaining label consistency inside each super-voxel. When adding one of our two key components individually (see the third and forth rows), results are improved on both datasets in each case, while the best results arise when using the full method with both key components.

4.4.2 Discussion on sample selection

We investigated sample selection via active learning further, to provide insights into why our method outperforms previous methods.

Figure 4 gives statistics on the classes of objects where the points were selected with the active learning in our method, compared to the method of Hou et al. We see that our method selects fewer samples on floors while selecting more samples on categories like wall, door, window, desk, counter, and sink, which leads to significant improvement of prediction accuracy for these categories. Our method even improves the prediction accuracy for floors as this class often gets

Table 3 Ablation studies of our method conducted with "20 points/scene" annotation. ScanNet-v2 is evaluated on the validation set while S3DIS is evaluated on Area 5. "Voting" indicates averaging the prediction of all the points inside of the same super-voxel during inference. "Self-train." refers to the self-training approach used to generate the pseudo label set to train the segmentation network. "Active learn." refers to the active learning method used to select the samples for annotation, which are propagated to constitute a per-point true label set

Component			mIoU (%)	
Voting	Self-train.	Active learn.	ScanNet	S3DIS
			59.2	39.5
✓			62.3	40.5
✓	✓		67.2	47.2
✓		✓	65.2	51.8
✓	✓	✓	**69.8**	**56.3**

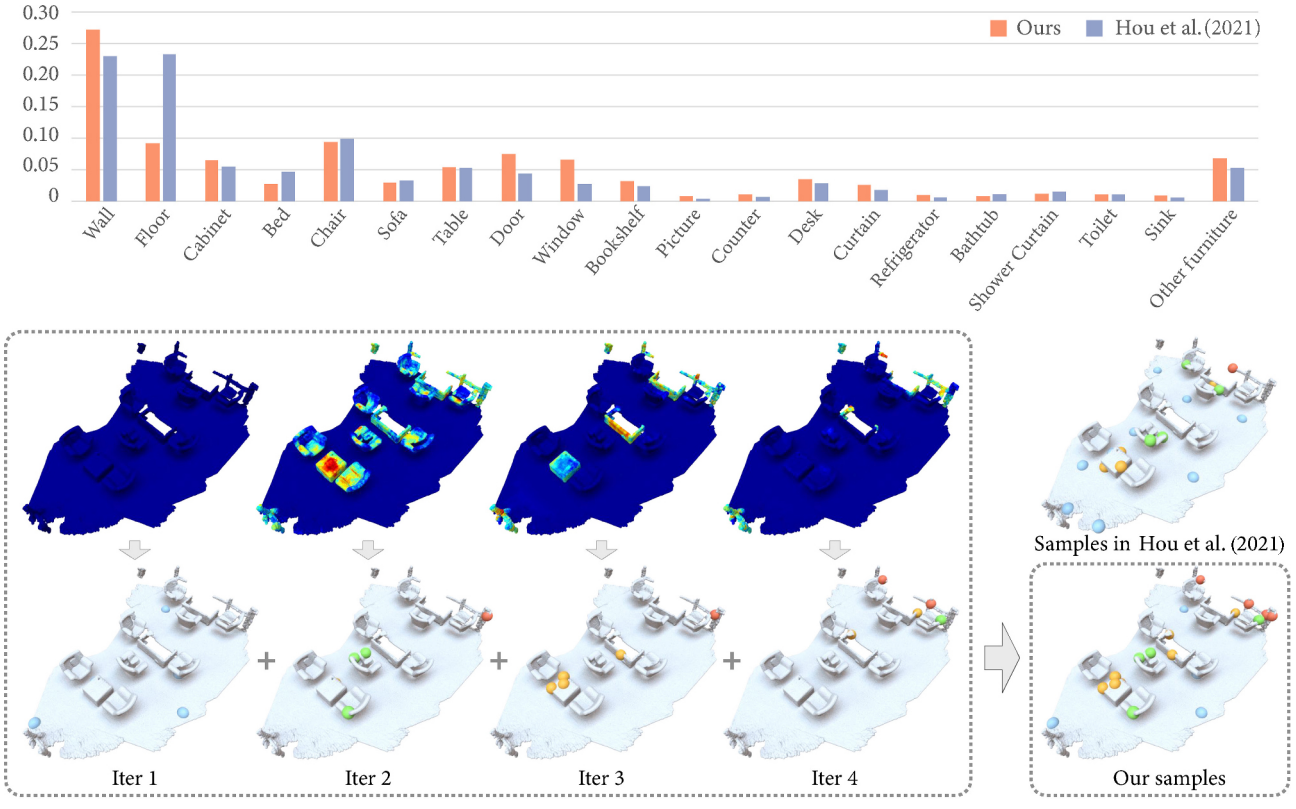


Fig. 4 Comparison of samples selected for annotation. Top: statistics on the classes of objects where points were selected with the active learning in our method, compared to Hou et al. Bottom: a visual comparison between the samples that the active learning selected based on uncertainty across four iterations and those selected at once based on a pre-trained model Hou et al. (top-right scene outside the dashed box).

misclassified as door in early-iteration results, while in the final results, there is less confusion with doors as our method selects more samples for doors.

We also show a visual comparison of selected samples in Fig. 4. For our method, we show the uncertainty map over the scene, which is used to guide the sample selection, together with the selected samples for each iteration. Compared to the samples selected at once with the pre-trained model of Hou et al., the samples we select are located on more complicated objects with more visual variability like chairs, rather than the floor.

Note that, if we have s scenes in the training set, there are two ways for active learning to sample the points. One way is to sample the same number of points in each scene, i.e., $n/(ks)$ per scene, and the other is to sample the points among all the scenes based on uncertainty only, which results in different numbers of sample points in different scenes. We tested these two different options and found the results to be similar. This is because both datasets have large scene variations, which leads to samples being evenly distributed over the scenes even if they are sampled over the entire dataset based on pointwise uncertainty. If another dataset with scenes similar to each other were given, we believe that the results from sampling over all the scenes would be better.

4.5 Results under "one thing one click" setting

We also tested our method under the "one thing one click" setting. The only change in our method is that, for sample selection during the active learning, we avoid objects that have been sampled before, selecting the sample with highest uncertainty among the remaining objects. In other words, our active learning chooses which object to sample as well as which point inside each object should be sampled in each iteration.

In more detail, we first computed the average number n_o of objects per scene for both datasets, which resulted in $n_o = 32$ for ScanNet-v2 and $n_o = 36$ for S3DIS. Here we set the number of iterations to be $k = 6$. For each of the first five iterations, we selected 6 samples from 6 different objects in the scene and set them as invalid for selection in the next iteration. In the final iteration, we selected one point with the

highest uncertainty from each remaining object to complete "one thing one click" sample selection.

Table 4 shows how the accuracy changes across iterations. For the S3DIS dataset, our method already obtains better results than Liu et al. when only 24 points were annotated in iteration 4. Figure 5 gives a visual comparison of a sample point selected by our method and the point randomly sampled as in Liu et al. We can see that our method learns to select points that are located in more important regions inside objects, which is more informative and leads to more accurate prediction results. By focusing on labeling more challenging regions, our method can correctly predict other unlabeled regions, while the random samples of Liu et al. may not be able to extract enough information to correctly predict the labels for the challenging regions.

One interesting result that we observed is that, for both the ScanNet-v2 and S3DIS datasets, after adding 24 points while constraining only one point per object, the results are worse than for our method under the 20 points/scene setting, where fewer annotations are given. To find the reason behind this behavior of the method, in Fig. 6, we give statistics of the number of points sampled on each object for our method under the 20 points/scene setting. We see in the results

Table 4 Evaluation of our method in the "one thing one click" setting, selecting one point per object during the active learning

Method	mIoU (%)	
	ScanNet	S3DIS
Liu et al.	70.5	50.1
Iter 1 (+6 points)	45.7	36.5
Iter 2 (+6 points)	62.5	45.8
Iter 3 (+6 points)	66.9	48.0
Iter 4 (+6 points)	68.7	51.2
Iter 5 (+6 points)	69.3	53.3
Iter 6 (+1 point for each remaining object)	**71.5**	**54.9**

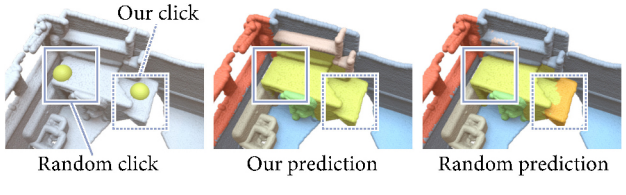


Fig. 5 Visual comparison of a sample point selected by our method and the point randomly sampled as in Liu et al., with the corresponding prediction results. We see that our sample is located on a more challenging region of the desk which leads to a more accurate prediction after training.

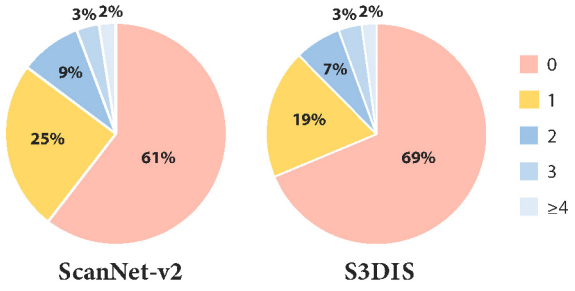


Fig. 6 Statistics on the number of points sampled on each object by our method under the 20 points/scene setting.

for ScanNet-v2 that, compared to the "one thing one click" setting where 100% of the objects get exactly one click, most objects (61%) in our setting do not get any samples, 14% of the objects get more than one click, and only 25% of the objects get exactly one click. We observed a similar distribution on S3DIS. This indicates that the "one thing one click" procedure may not be a good way to sample points for semantic segmentation. Complicated objects may need more sample points than others, while objects in simpler categories do not require annotating for each scene.

5 Limitations and future work

Currently, in each iteration, self-training uses the labels collected from the active learning and the pseudo labels only once. However, it is possible to use the active learning as an outer loop and self-training as an inner loop of the method, to run more iterations of self-training for each set of annotated points. This may lead to better results at the expense of increasing the training time. It is also possible to use a pre-trained model as in Hou et al. to obtain a better set of initial samples for annotation. Theoretically, the work of Liu et al. requires extra time to identify each individual object when annotating the same number of points as our method, while the drawback of our point selection based on active learning is that points need to be annotated in several iterations instead of once as in Liu et al. Although we believe that annotations can be collected through crowdsourcing on the Internet and thus no extra user effort is needed as users do not have to wait in front of the computer, this would cause information delay and would become a problem when collection is conducted in person. While we adopt the same supervoxel clustering methods as in Liu et al. for a fair comparison, it is worth exploring

other more advanced methods such those of as Huang et al. and Lin et al. [39] to further boost results.

6 Conclusions

We have introduced a weakly supervised method for semantic segmentation of 3D scenes, which combines an active learning component that selects the most effective points to be annotated by users, and a self-training approach that makes efficient use of the user labels. We have shown that our method leads to improvements of 11% and 6% mIoU over previous works on well-known datasets, while using the same number of or fewer user annotations. Our method is also competitive with fully supervised methods.

Appendix

In this appendix, we first present more details of our ActiveST framework in Appendix A. Then, in Appendix B, we report detailed benchmark results on the ScanNet-v2 test set with per-category results of our method compared to other weakly supervised methods in the most challenging setting with only 20 points annotated in each scene. Finally, in Appendix C, we show more results on both ScanNet-v2 and S3DIS datasets with various numbers of annotated points.

A Details of our ActiveST framework

We present the training procedure for our proposed ActiveST framework in Algorithm 1. Figure 7 shows the 3D U-Net architecture used as the backbone. This architecture was proposed by Choy et al. for semantic segmentation, and contains four blocks for encoding and four blocks for decoding. For each block, we show the output dimension D and number of convolution layers N.

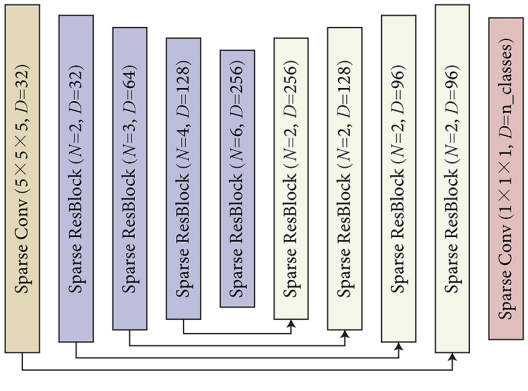


Fig. 7 3D U-Net architecture of Choy et al. used as our backbone.

Algorithm 1 Training procedure of our ActiveST framework

Input: Total amount of user annotations n, number of iterations k, super-voxel partition;

1: Random sample $m = n/k$ points to annotate, and propagate the labels in super-voxels to get T;
2: Train a network θ using T;
3: **for** $i = 1$ to $k - 1$ **do**
4: Use θ to generate pseudo-labels and compute uncertainty and confidence;
5: Select m points to annotate according to the uncertainty distribution, and propagate the labels in super-voxels to get T_i;
6: Update true labels $T = T \cup T_i$;
7: Select pseudo-labels with high confidence on unlabeled points to get P;
8: Train a new network θ_i using T and P with the softmax cross-entropy loss;
9: $\theta \leftarrow \theta_i$
10: **end for**

Output: Segmentation network θ.

B Per-category results on ScanNet-v2

In Table 5, we compare detailed per-category results for our method to those for other weakly supervised methods in the "data efficient" setting with 20 labeled points per scene as supplement. We can see that our method achieves significant improvements on categories such as bathtub, counter, cabinet, sink, and window. We believe that the reason is that our method is able to sample more points on more challenging categories via active learning, compared to the default set of sampled points used in other methods.

We also note that the only category for which our method gets worse results than the method of Liu et al. is the picture category. We believe that this is because our method puts more samples on other categories that lead to more confusion in the output prediction, leading to fewer annotations for the picture category. As we show in Appendix C, when given a larger annotation budget, more points are sampled on the picture category and thus the results are highly improved.

C Results on ScanNet-v2 and S3DIS

In this section, we show the results of our method on both ScanNet-v2 and S3DIS datasets with an increasing number of annotated points, using 20, 50, 100, and 200 points/scene.

We first report the results on the ScanNet-v2 test

Table 5 Per-category performance of our method compared to other weakly supervised methods on the ScanNet-v2 data-efficient benchmark (20 labeled points per scene for training)

Method	mIoU (%)	bath.	bed	bosh.	cabn.	chair	coun.	curt.	desk	door	floor	othfur.	pict.	refrig.	show.	sink	sofa	table	toilet	wall	wind.
Hou et al.	53.1	65.9	63.8	57.8	41.7	77.5	25.4	53.7	39.6	43.9	93.9	28.4	8.3	41.4	59.9	48.8	69.8	44.4	78.5	74.7	44.0
Xie et al.	55.0	73.5	67.6	60.1	47.5	79.4	28.8	62.1	37.8	43.0	94.0	30.3	8.9	37.9	58.0	53.1	68.9	42.2	85.2	75.8	46.8
Liu et al.	59.4	75.6	72.2	49.4	54.6	79.5	37.1	72.5	55.9	48.8	95.7	36.7	**26.1**	54.7	57.5	22.5	67.1	54.3	90.4	82.6	55.7
Ours	**70.3**	**97.7**	**77.6**	**65.7**	**70.7**	**87.4**	**54.1**	**74.4**	**60.5**	**61.0**	**96.8**	**44.2**	12.6	**70.5**	**78.5**	**74.2**	**79.1**	**58.6**	**94.0**	**83.9**	**64.5**

set. As Table 6 shows, using an increasing number of sample points, results consistently improve for all methods; our method always achieves the highest mIoU under all settings. We can see that our method surpasses all other weakly supervised approaches in the same setting by a large margin and sets a new state-of-the-art doe the Scannet-v2 "data-efficient" challenge. Note that, when only using 20 labeled points per scene, our method even beats the most competitive weakly supervised approach Liu et al. which uses 10 times as many points (200 labeled points per scene) by nearly 1% mIoU.

Table 6 Comparison of our method (ActiveST) to other methods under different limited point annotations, where "pts" means points/scene. We report mIoU (%) on Scanet-v2 test set

Method	20 pts	50 pts	100 pts	200 pts
Hou et al.	53.1	61.2	64.4	66.5
Xie et al.	55.0	61.4	63.5	65.3
Liu et al.	59.4	64.2	67.0	69.4
ActiveST (ours)	**70.3**	**72.5**	**73.5**	**74.8**

With greater annotation budget, we observe that our method tends to select more points on the picture category, the category with the worst results under the 20 labeled points per scene setting (see Appendix B): the mIoU of the picture category increases by 25.6% going from 20 to 200 points per scene. The percentage of points sampled on the window category also increases, with an mIoU improvement of 5.6% from 20 to 200 points. Figure 8 compares samples selected by our method and Hou et al. under different settings. It can be observed that our method selects more samples on picture and window categories instead of floor in each setting, leading to a great improvement in the final results.

We also notice that although the sample proportion of some categories such as toilet, cabinet, and curtain become smaller, mIoU for those categories is not sacrificed as our model is less likely to misclassify objects in those categories as wall due to increasing samples for the wall category. We also report our

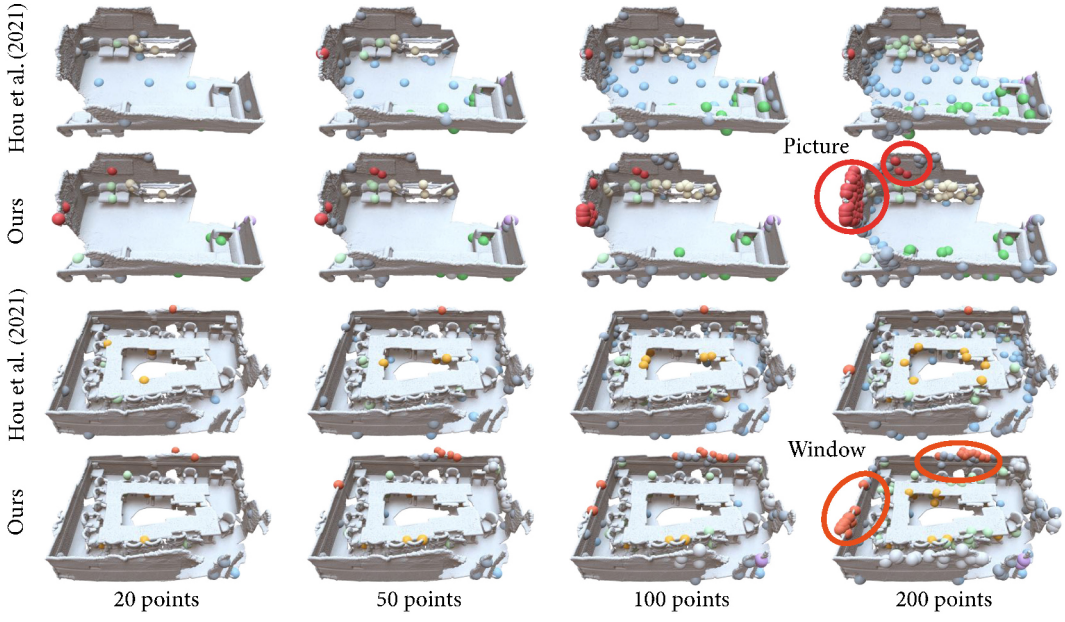


Fig. 8 Comparison of samples selected by our method and the method of Hou et al. with different annotation budget.

results on S3DIS under these various settings in Table 7. With more points selected by our method, the gap between our method and the fully supervised baseline is further reduced.

Table 7 Comparison of our method (ActiveST) to our fully-supervised baseline under different limited point annotations. We report mIoU (%) on Area-5 of S3DIS

Method	Supervision	mIoU (%)
ActiveST (ours)	20 pts	56.3
ActiveST (ours)	50 pts	57.8
ActiveST (ours)	100 pts	59.5
ActiveST (ours)	200 pts	62.1
Our fully-sup baseline	100%	66.3

Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work was supported by Guangdong Natural Science Foundation (2021B1515020085), Shenzhen Science and Technology Program (RCYX20210609103121030), National Natural Science Foundation of China (62322207, 61872250, U2001206, U21B2023), Department of Education of Guangdong Province Innovation Team (2022KCXTD025), Shenzhen Science and Technology Innovation Program (JCYJ20210324120213036), the Natural Sciences and Engineering Research Council of Canada (NSERC), and Guangdong Laboratory of Artificial Intelligence and Digital Economy (ShenZhen).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Qi, C. R.; Yi, L.; Su, H.; Guibas, L. J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 5105–5114, 2017.

[2] Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution on X-transformed points. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 828–838, 2018.

[3] Thomas, H.; Qi, C. R.; Deschaud, J. E.; Marcotegui, B.; Goulette, F.; Guibas, L. KPConv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 6410–6419, 2019.

[4] Han, L.; Zheng, T.; Xu, L.; Fang, L. OccuSeg: Occupancy-aware 3D instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2937–2946, 2020.

[5] Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; Niessner, M. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2432–2443, 2017.

[6] Armeni, I.; Sax, S.; Zamir, A. R.; Savarese, S. Joint 2D-3D-semantic data for indoor scene understanding. *arXiv preprint* arXiv:1702.01105, 2017.

[7] Wei, J. C.; Lin, G. S.; Yap, K. H.; Hung, T. Y.; Xie, L. H. Multi-path region mining for weakly supervised 3D semantic segmentation on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4383–4392, 2020.

[8] Xu, X.; Lee, G. H. Weakly supervised semantic point cloud segmentation: Towards 10× fewer labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 13703–13712, 2020.

[9] Gadelha, M.; RoyChowdhury, A.; Sharma, G.; Kalogerakis, E.; Cao, L. L.; Learned-Miller, E.; Wang, R.; Maji, S. Label-efficient learning on point clouds using approximate convex decompositions. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12355*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 473–491, 2020.

[10] Xie, S. N.; Gu, J. T.; Guo, D. M.; Qi, C. R.; Guibas, L.; Litany, O. PointContrast: Unsupervised pre-training for 3D point cloud understanding. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12348*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 574–591, 2020.

[11] Jiang, L.; Shi, S. S.; Tian, Z. T.; Lai, X.; Liu, S.; Fu, C. W.; Jia, J. Y. Guided point contrastive learning for semi-supervised point cloud semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 6403–6412, 2021.

[12] Hou, J.; Graham, B.; Niesner, M.; Xie, S. N. Exploring data-efficient 3D scene understanding with contrastive scene contexts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 15582–15592, 2021.

[13] Choy, C.; Gwak, J.; Savarese, S. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on

Computer Vision and Pattern Recognition, 3070–3079, 2019.

[14] Liu, Z. Z.; Qi, X. J.; Fu, C. W. One thing one click: A self-training approach for weakly supervised 3D semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1726–1736, 2021.

[15] Hu, R. Z.; Wen, C.; Van Kaick, O.; Chen, L. M.; Lin, D.; Cohen-Or, D.; Huang, H. Semantic object reconstruction via casual handheld scanning. *ACM Transactions on Graphics* Vol. 37, No. 6, Article No. 219, 2018.

[16] Wu, T. H.; Liu, Y. C.; Huang, Y. K.; Lee, H. Y.; Su, H. T.; Huang, P. C.; Hsu, W. H. ReDAL: Region-based and diversity-aware active learning for point cloud semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 15490–15499, 2021.

[17] Shi, X.; Xu, X.; Chen, K.; Cai, L.; Foo, C. S.; Jia, K. Label-efficient point cloud semantic segmentation: An active learning approach. *arXiv preprint* arXiv:2101.06931, 2021.

[18] Wu, Z. R.; Song, S. R.; Khosla, A.; Yu, F.; Zhang, L. G.; Tang, X. O.; Xiao, J. X. 3D ShapeNets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1912–1920, 2015.

[19] Charles, R. Q.; Hao, S.; Mo, K. C.; Guibas, L. J. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 77–85, 2017.

[20] Wu, W. X.; Qi, Z. A.; Li, F. X. PointConv: Deep convolutional networks on 3D point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern, 9613–9622, 2019.

[21] Komarichev, A.; Zhong, Z. C.; Hua, J. A-CNN: Annularly convolutional neural networks on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7413–7422, 2019.

[22] Su, H.; Jampani, V.; Sun, D. Q.; Maji, S.; Kalogerakis, E.; Yang, M. H.; Kautz, J. SPLATNet: Sparse lattice networks for point cloud processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2530–2539, 2018.

[23] Liu, Z.; Tang, H.; Lin, Y.; Han, S. Point-voxel CNN for efficient 3D deep learning. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Article No. 87, 965–975, 2019.

[24] Ye, X. Q.; Li, J. M.; Huang, H. X.; Du, L.; Zhang, X. L. 3D recurrent neural networks with context fusion for point cloud semantic segmentation. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11211*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 415–430, 2018.

[25] Guo, M. H.; Cai, J. X.; Liu, Z. N.; Mu, T. J.; Martin, R. R.; Hu, S. M. PCT: Point cloud transformer. *Computational Visual Media* Vol. 7, No. 2, 187–199, 2021.

[26] Peng, H. T.; Zhou, B.; Yin, L. Y.; Guo, K.; Zhao, Q. P. Semantic part segmentation of single-view point cloud. *Science China Information Sciences* Vol. 63, No. 12, 224101, 2020.

[27] Kundu, A.; Yin, X. Q.; Fathi, A.; Ross, D.; Brewington, B.; Funkhouser, T.; Pantofaru, C. Virtual multi-view fusion for 3D semantic segmentation. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12369*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 518–535, 2020.

[28] Dai, A.; Nießner, M. 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11214*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 458–474, 2018.

[29] Graham, B.; Engelcke, M.; van der Maaten, L. 3D semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9224–9232, 2018.

[30] Landrieu, L.; Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4558–4567, 2018.

[31] Tatarchenko, M.; Park, J.; Koltun, V.; Zhou, Q. Y. Tangent convolutions for dense prediction in 3D. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3887–3896, 2018.

[32] Huang, S. S.; Ma, Z. Y.; Mu, T. J.; Fu, H. B.; Hu, S. M. Supervoxel convolution for online 3D semantic segmentation. *ACM Transactions on Graphics* Vol. 40, No. 3, Article No. 34, 2021.

[33] Zhang, J. Z.; Zhu, C. Y.; Zheng, L. T.; Xu, K. Fusion-aware point convolution for online semantic 3D scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4533–4542, 2020.

[34] Cheng, M. M.; Hui, L.; Xie, J.; Yang, J. SSPC-net: Semi-supervised semantic 3D point cloud segmentation

network. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 35, No. 2, 1140–1147, 2021.

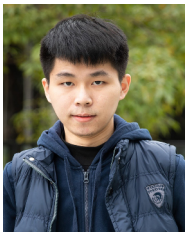
[35] Zhang, Z. W.; Girdhar, R.; Joulin, A.; Misra, I. Self-supervised pretraining of 3D features on any point-cloud. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 10232–10243, 2021.

[36] Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I. C.; Yan, M. Y.; Su, H.; Lu, C. W.; Huang, Q. X.; Sheffer, A.; Guibas, L. A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics* Vol. 35, No. 6, Article No. 210, 2016.

[37] Rizve, M. N.; Duarte, K.; Rawat, Y. S.; Shah, M. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint* arXiv:2101.06329, 2021.

[38] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga L.; et al. PyTorch: An imperative style, high-performance deep learning library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Article No. 721, 8026–8037, 2019.

[39] Lin, Y. B.; Wang, C.; Zhai, D. W.; Li, W.; Li, J. Toward better boundary preserved supervoxel segmentation for 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* Vol. 143, 39–47, 2018.



Gengxin Liu is currently pursuing a master degree in the College of Computer Science & Software Engineering, Shenzhen University under the supervision of Dr. Ruizhen Hu. His research interests include machine learning and point cloud processing.

Oliver van Kaick is an associate professor in the School of Computer Science at Carleton University, Ottawa, Canada. He received his Ph.D. degree from the School of Computing Science at Simon Fraser University (SFU). He was then a postdoctoral researcher at SFU and Tel Aviv University. His research concentrates on shape analysis and geometric modeling.

Hui Huang is a Distinguished TFA Professor of Shenzhen University, where she directs the Visual Computing Research Center. She received her Ph.D. degree in applied math from the University of British Columbia in 2008. Her research interests span computer graphics, vision, and visualization. She is currently a Senior Member of IEEE/ACM/CSIG, a Distinguished Member of CCF, and is on the editorial boards of *ACM Trans. Graphics* and *IEEE Trans. Visualization and Computer Graphics*.

Ruizhen Hu is an associate professor at Shenzhen University. She received her Ph.D. degree from the Department of Mathematics, Zhejiang University. Before that, she spent two years visiting Simon Fraser University, Canada. Her research interests are in computer graphics, with a recent focus on applying machine learning to advance the understanding and generative modeling of visual data including 3D shapes and indoor scenes. She is an editorial board member of *The Visual Computer* and *IEEE CG&A*.