SL3D: Self-supervised-Self-labeled 3D Recognition

Fernando Julio Cendra
The University of Hong Kong
&
TCL AI Lab
fcendra@connect.hku.hk

Lan Ma
TCL AI Lab
rubyma@tcl.com

Jiajun Shen
TCL AI Lab
shenjiajun90@gmail.com

Xiaojuan Qi
The University of Hong Kong
xjqi@eee.hku.hk

Abstract

Deep learning has attained remarkable success in many 3D visual recognition tasks, including shape classification, object detection, and semantic segmentation. However, many of these results rely on manually collecting densely annotated real-world 3D data, which is highly time-consuming and expensive to obtain, limiting the scalability of 3D recognition tasks. Thus, we study unsupervised 3D recognition and propose a **S**elf-supervised-Self-**L**abeled **3D** Recognition (SL3D) framework. SL3D simultaneously solves two coupled objectives, *i.e.,* clustering and learning feature representation to generate pseudo-labeled data for unsupervised 3D recognition. SL3D is a generic framework and can be applied to solve different 3D recognition tasks, including classification, object detection, and semantic segmentation. Extensive experiments demonstrate its effectiveness. Code is available at `https://github.com/fcendra/sl3d`.

1 Introduction

3D object recognition is a fundamental problem in computer vision that aims to process given visual data to generate high-level understandings with many applications in robotics, autonomous driving, and virtual reality. Some representative recognition tasks include object classification, detection, and semantic segmentation. Although 3D data acquisition has been convenient and inexpensive with the advancement of 3D sensing technologies, annotating 3D labels is still laborious, inconvenient, and time-consuming, which limits the scalability of many recognition tasks. For example, on average, it takes 22.3 minutes to annotate the ground-truth data for a single 3D indoor scene data sample on ScanNet [1] dataset, and nearly 500 annotators were involved in constructing the dataset. This problem directly causes a hindrance towards unleashing the full potential of deep learning methods, as the key strength of deep learning methods is their ability to learn hierarchical features from a large amount of annotated training data. The unavailability of large-scale labeled data has been one of the most significant factors preventing the community from scaling 3D recognition tasks. We review the related works in greater detail in Sec. B.

To this end, we explore the potential of unsupervised 3D recognition, which does not require annotations for model training and thus bypasses the limitations of data annotations. To our best knowledge, this has rarely been explored in the 3D domain. We propose the SL3D learning paradigm, which tackles 3D recognition problem by simultaneously self-labeling and learning feature representation from unlabeled object-level point cloud data. Inspired by SeLa [2], SL3D generates pseudo labels by simultaneously clustering with an equipartition constraint (i.e., each cluster is assigned with an approximately equal number of samples) and learning feature representations from clustered data.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Clustering and representation learning is conducted iteratively to finally deliver high-quality pseudo labeled data. Furthermore, these pseudo labeled data obtained from SL3D can be used as supervision for several downstream tasks such as 3D object classification, 3D object detection, and 3D semantic segmentation to achieve unsupervised 3D recognition. As object detection and semantic segmentation tasks deal with scene-level point cloud data, a selective search algorithm is used to generate bounding box proposals from 3D indoor dataset and transform them into object-level point cloud.

We conduct extensive experiments on ModelNet40 dataset [3] for the object classification task, and ScanNet [1] dataset for both object detection and semantic segmentation tasks. The experimental results show that our method achieves promising results for unsupervised 3D object detection and 3D semantic segmentation. Moreover, through extensive experiments, we show that our Unsupervised 3D object detection achieves on-par results compared with the base benchmark model for Weakly-supervised 3D object detection. Our contributions can be summarized as below:

- We propose SL3D for unsupervised 3D recognition, a generic framework for 3D object classification, semantic segmentation, and object detection.
- We conduct extensive experiments to verify the effectiveness of our framework on three challenging 3D recognition tasks.

The rest of the paper is organized as follows. Section 2 gives an overview of our method. Section 3 reports the experimental results of our SL3D model on downstream tasks. Finally, in section 4, the paper ends with a conclusion.

2 Method

The core task of our framework, summarized in Sec. A, is to generate pseudo labels from given point cloud data which can be used as training data for downstream 3D object recognition tasks, including object classification, semantic segmentation, and object detection. To achieve our goal, we design a simultaneous self-supervised-self-labeled (SL3D) method that generates labels through clustering from sets of object-level point clouds $\{P_j^O \mid j = 1, \dots, |P^O|\}$, where P_j^O is a vector consists of (x, y, z) coordinates representing object-level point cloud j. Given scene-level point cloud data $\{P_i^S \mid i = 1, \dots, m\}$ as the input, a bottom-up 3D geometric selective search object proposal algorithm [4] is employed to generate sets of P_j^O for each P_i^S, where i corresponds to the scene index. Afterward, the scene point cloud is represented as a set of object-level proposals. In the following, we will first elaborate the SL3D model for generating high-quality pseudo labels. Then, we will explain how the pseudo labeled data are used for solving downstream tasks.

2.1 Simultaneous self-supervised-Self-labeled model

Given a set of object-level point clouds X = $\{x \mid \forall x \in P^O\}$ which are either data samples from an object-level classification dataset such as ModelNet40 [3] dataset or proposals generated from a scene-level point cloud dataset such as ScanNet [1] dataset, we develop our self-supervised-self-labeled (SL3D) framework inspired by a 2D-based self-supervised model proposed by Asano *et al.*[2]. The key idea behind this method is to simultaneously train a feature representation learning model that learns from a joint distribution generated by the clustering model. Therefore, the model is divided into two coupled parts *i.e.,* **1. Clustering model:** Given features from a network, a clustering model is developed to assign data to a pseudo cluster with an equipartition constraint to ensure the data distribute equally amongst the pseudo classes. **2. Feature representation learning:** The network is supervised by minimizing the cross-entropy loss between pseudo label distribution Q and the distribution P obtained from the model predictions. The two models work collaboratively to update the backbone network $f(\cdot)$ which can, in turn, generate better pseudo labels.

Clustering model: Given joint probability distribution P obtained from Eq.(1) where each element represents the probability of a sample belonging to a class, a naive approach would be to assign a data sample to the class with the largest probability and using it as supervision for feature representation learning. However, this would lead to a degeneracy solution as all the samples X have the potential to be clustered into one pseudo class.

$$P_{y_i} = \frac{1}{N}\text{softmax}(h \circ f(x_i)) \tag{1}$$

To avoid this undesirable situation, we adopt the approach proposed by Asano *et al.*[2], which employs an equipartition constraint to ensure X to be assigned uniformly to each pseudo class, i.e., balanced pseudo clusters. Initially, we set P to be a $K \times N$ joint probabilities matrix obtained following Eq.(1), where N is the number of object-level data samples in X, K is the number of pseudo classes (groups), h is the classifier head with a length of K, and softmax$(\cdot)$ is the normalization operation. Similarly, we also initialize label assignment matrix Q to be $K \times N$. Each element Q_{ij} will be used to represent the pseudo probability of a sample j belonging to a group i following the equipartition constraint.

Feature representation learning model: Given probability distribution P label assignment matrix Q obtained from the clustering model, we train a representation learning model by minimizing the cross-entropy loss CE$(\cdot, \cdot)$ between Q and P. Thus, the objective function of this model can be written as $\min\langle Q, -\log P\rangle$, where $\langle\cdot\rangle$ is the Frobenius dot product operation between two matrices. Finally, a fast version of Sinkhorn-Knopp algorithm [5] is used to optimize this objective function efficiently. Thus, we see that by clustering, and learning representations simultaneously, minimizing the objective function is the same way as maximizing the mutual information between the label and data index i while still maintaining equipartition among the N data. To obtain the pseudo labels Y for further use in 3D recognition tasks, we can simply take the argument that gives the maximum value for every data index in table Q, this can be formulated as $Y = \{y \mid \forall y \in \text{argmax}(Q_{*,i}), \forall i \in (1, \ldots, N)\}$.

2.2 Downstream tasks

As the downstream task methods are not the core components of SL3D, each task adopts the model from existing 3D object recognition works. As each downstream task requires different types of input training data, another pre-processing task is needed to transform SL3D dataset for each 3D recognition task. The 3D object classification task requires object-level point cloud data and its corresponding class label. The class labels can be obtained from pseudo label generated by SL3D. For classifier model, we use PointNet++ [6] as our baseline model. For 3D indoor scene recognition, we evaluate both 3D object detection and 3D semantic segmentation. 3D object detection task requires the semantic labels and bounding box location of each objects, we can directly get the semantic labels from SL3D pseudo labels and obtain the bounding box location from the proposals generated by GSS. After obtaining the object detection dataset from SL3D model, we train VoteNet [7] as our object detection model. For 3D semantic segmentation task, as SL3D does not have any point-wise labels, we use the obtained proposal boxes to map and assign all points inside the box with the corresponding SL3D pseudo label. After obtaining the 3D semantic segmentation dataset from SL3D, we use PointNet++ [6] semantic segmentation model and train it with our pseudo labeled data.

3 Experiment

We evaluate and compare SL3D framework with the state-of-the-art weakly supervised and supervised methods on three 3D object recognition tasks,i.e., 3D object classification on ModelNet40 [3] dataset, 3D object detection and semantic segmentation benchmarks on ScanNet [1] dataset. We also present our implementation details in Sec. C, and ablation experiments in Sec. D.

3.1 Dataset

For 3D classification tasks, we use the ModelNet40 dataset. It contains $12,311$ object-level point cloud CAD models of 40 categories. The dataset is divided into a training set ($9,843$ objects) and a testing set ($2,468$ objects). For 3D object detection and 3D semantic segmentation tasks, we use real 3D scene point cloud data, ScanNet [1] dataset. ScanNet consists of 1613 scene-level indoor scene point cloud data annotated with 20 semantic classes. The dataset is divided into training (1201 scenes), validation (312 scenes), and testing (100 scenes) sets.

3.2 Main Results

Here we present our unsupervised 3D recognition model performance for classification, object detection, and semantic segmentation tasks. We use mean average precision (mAP@0.25) for detection task. Moreover, for segmentation task, we use mean intersection over union (mIoU) to measure the similarity of the point-level semantic ground truth with our SL3D prediction result. As there are no prior works in unsupervised 3D indoor-scene recognition, we show other learning methods, i.e., weakly-supervised and supervised methods, as a reference. The evaluation results are shown in Table 1 for classification, Table 2 for object detection, and Table 3 for semantic segmentation. We define our model settings to be $SL3D_K^{\{\dagger,\ddagger\}}$, where † denotes SL3D with PointNet++ backbone, ‡ denotes SL3D with Point Transformer backbone, and K denotes the number of pseudo classes.

Unsupervised 3D object classification on ModelNet40 dataset: As Table 1 shows, our model performs better with the increased number of pseudo classes. Its accuracy gains around 9.7% (± 2.9) improvement for every doubling of pseudo class number. Moreover, compared with the baseline and SOTA supervised classification models, our best model (with an accuracy of 77.2%) achieves accuracy differences of 14.7 and 17.3, respectively. **Unsupervised 3D object detection on ScanNet dataset:** As shown in Table 2, our model obtains decent results in this unsupervised setting. It is reasonable that our model underperforms compared with other models in the weakly supervised or the strongly supervised setting. Still, our model (with Point Transformer as the backbone architecture and 400 pseudo classes) achieves competitive results compared to MIL-det [4], a weakly-supervised object detection-based model. We also evaluate our model under the same configuration on SL3D object detection validation set, i.e., the labels are obtained from SL3D model, and the model achieves an mAP of 19.1%. This performance improvement implies that there is a discrepancy between object groups discovered by SL3D and the ground-truth annotated categories, which leaves room for further improvements. **Unsupervised 3D semantic segmentation on ScanNet dataset:** As shown in Table 3, we show that our unsupervised semantic segmentation model performs best with Point Transformer backbone and 400 pseudo classes and achieves 10.4% of $mIoU$. We observe that increasing the number of pseudo classes improves the overall semantic segmentation results.

Table 1: 3D object classification results on ModelNet40 dataset from different level of supervision.

No.	Method	No. of class obtained	Overall Acc($\%$)
Unsupervised methods			
1	$SL3D^{\ddagger}_{100}$ (ours)	26	38.7
2	$SL3D^{\ddagger}_{200}$ (ours)	33	48.9
3	$SL3D^{\ddagger}_{400}$ (ours)	38	59.8
4	$SL3D^{\ddagger}_{800}$ (ours)	40	72.5
4	$SL3D^{\ddagger}_{1600}$ (ours)	40	**77.2**
Supervised methods			
5	PointNet++ [6]	40	91.9
6	PointMLP [8]	40	94.5

Table 2: 3D object detection results (mAP@0.25) task on ScanNet validation set from different level of supervision.

No.	Method	No. of class obtained	Split	mAP@0.25
Unsupervised methods				
1	$SL3D^{\dagger}_{50}$ (ours)	7	Val(SL3D)/Val	18.6/4.6
2	$SL3D^{\ddagger}_{100}$ (ours)	12	Val(SL3D)/Val	17.8/7.6
3	$SL3D^{\ddagger}_{200}$ (ours)	15	Val(SL3D)/Val	20.3/7.9
4	$SL3D^{\ddagger}_{400}$ (ours)	18	Val(SL3D)/Val	19.1/**9.3**
Weakly-supervised methods				
5	MIL-det [4]	18	Val	9.6
6	WyPR [4]	18	Val	18.3
7	VoteNet + WS3D [9]	18	Val	18.4
8	WyPR + prior [4]	18	Val	19.7
9	VoteNet + BR_P [10]	18	Val	31.2
Supervised methods				
10	F-PointNet [11]	18	Val	19.8
11	GSPN [12]	18	Val	30.6
12	VoteNet [7]	18	Val	58.6
13	RBGNet [13]	18	Val	70.6

Table 3: 3D semantic segmentation results on ScanNet dataset from different level of supervision. Val(SL3D) denotes the SL3D semantic segmentation validation set, and Val(sub-cloud) denotes the PCAM [14] and MPRM [15] validation sets with sub-cloud supervision.

No.	Method	No. of class obtained	Split	mIoU
Unsupervised methods				
1	$SL3D^{\dagger}_{50}$ (ours)	9	Train/Val(SL3D)/Test	60.2/32.9/5.8
2	$SL3D^{\ddagger}_{100}$ (ours)	14	Train/Val(SL3D)/Test	57.3/26.6/8.4
3	$SL3D^{\dagger}_{400}$ (ours)	18	Train/Val(SL3D)/Test	56.1/28.5/8.5
4	$SL3D^{\ddagger}_{400}$ (ours)	20	Train/Val(SL3D)/Test	55.1/25.3/9.2
5	$SL3D^{\ddagger}_{800}$ (ours)	20	Train/Val(SL3D)/Test	53.6/24.7/10.5
Weakly-supervised methods				
6	PCAM [14]	20	Train/Val(sub-cloud)	22.1/28.1
7	MPRM [15]	20	Train/Val(sub-cloud)	24.4/41.0
8	WyPR [4]	20	Train/Val/Test	30.7/29.6/24.0
9	WyPR + prior [4]	20	Val	31.1
Supervised methods				
10	PointNet++ [6]	20	Test	33.9

4 Conclusions

SL3D framework has been proposed in this paper. Given a 3D object-level point cloud data (optionally for scene-level point cloud data, a geometric selective search algorithm is used to create object-level point cloud data), These set of point clouds are then fetched into the simultaneous self-supervised-self-labeled learning model to acquire pseudo labels. The pseudo labeled data are further used to supervise different 3D recognition model training including classification, object detection, and semantic segmentation. Our experimental results indicate that this model fulfills our main research aims, and SL3D can be a good research baseline for unsupervised 3D recognition. Moreover, SL3D generates good clusters of pseudo labels, and its pretrained weights can improve supervised learning via transfer learning. We hope this work can inspire others to do further research in this area.

References

[1] Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE. (2017)

[2] Asano, Y.M., Rupprecht, C., Vedaldi, A.: Self-labelling via simultaneous clustering and representation learning. In: International Conference on Learning Representations (ICLR). (2020)

[3] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 1912–1920

[4] Ren, Z., Misra, I., Schwing, A.G., Girdhar, R.: 3d spatial recognition without spatially labeled 3d. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2021)

[5] Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems **26** (2013)

[6] Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017)

[7] Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE International Conference on Computer Vision. (2019)

[8] Ma, X., Qin, C., You, H., Ran, H., Fu, Y.: Rethinking network design and local geometry in point cloud: A simple residual mlp framework. arXiv preprint arXiv:2202.07123 (2022)

[9] Meng, Q., Wang, W., Zhou, T., Shen, J., Gool, L.V., Dai, D.: Weakly supervised 3d object detection from lidar point cloud. In: European Conference on Computer Vision, Springer (2020) 515–531

[10] Xu, X., Wang, Y., Zheng, Y., Rao, Y., Zhou, J., Lu, J.: Back to reality: Weakly-supervised 3d object detection with shape-guided label enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 8438–8447

[11] Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 918–927

[12] Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J.: Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 3947–3956

[13] Wang, H., Shi, S., Yang, Z., Fang, R., Qian, Q., Li, H., Schiele, B., Wang, L.: Rbgnet: Ray-based grouping for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 1110–1119

[14] McEver, R.A., Manjunath, B.: Pcams: Weakly supervised semantic segmentation using point supervision. arXiv preprint arXiv:2007.05615 (2020)

[15] Wei, J., Lin, G., Yap, K.H., Hung, T.Y., Xie, L.: 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. (2020) 4384–4393

[16] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in Neural Information Processing Systems **33** (2020) 9912–9924

[17] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.B.: Momentum contrast for unsupervised visual representation learning. CoRR **abs/1911.05722** (2019)

[18] Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: European Conference on Computer Vision. (2018)

[19] Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. CoRR **abs/1505.05192** (2015)

[20] Noroozi, M., Vinjimoor, A., Favaro, P., Pirsiavash, H.: Boosting self-supervised learning via knowledge transfer. CoRR **abs/1805.00385** (2018)

[21] Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. CoRR **abs/1603.08511** (2016)

[22] Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding (2017)

[23] Jenni, S., Favaro, P.: Self-supervised feature learning by learning to spot artifacts (2018)

[24] Noroozi, M., Pirsiavash, H., Favaro, P.: Representation learning by learning to count (2017)

[25] Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)

[26] Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations (2018)

[27] Ji, X., Henriques, J.F., Vedaldi, A.: Invariant information clustering for unsupervised image classification and segmentation (2019)

[28] Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L.J., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding (2020)

[29] Jiang, L., Shi, S., Tian, Z., Lai, X., Liu, S., Fu, C.W., Jia, J.: Guided point contrastive learning for semi-supervised point cloud semantic segmentation (2021)

[30] Sauder, J., Sievers, B.: Self-supervised deep learning on point clouds by reconstructing space (2019)

[31] Poursaeed, O., Jiang, T., Qiao, H., Xu, N., Kim, V.G.: Self-supervised learning of point clouds via orientation estimation (2020)

[32] Thabet, A., Alwassel, H., Ghanem, B.: Mortonnet: Self-supervised learning of local features in 3d point clouds (2019)

[33] Afham, M., Dissanayake, I., Dissanayake, D., Dharmasiri, A., Thilakarathna, K., Rodrigo, R.: Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding (2022)

[34] Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE international conference on computer vision. (2015) 945–953

[35] Yu, T., Meng, J., Yuan, J.: Multi-view harmonized bilinear network for 3d object recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 186–194

[36] Yavartanoo, M., Kim, E.Y., Lee, K.M.: Spnet: Deep 3d object classification and retrieval using stereographic projection. In: Asian conference on computer vision, Springer (2018) 691–706

[37] Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE (2015) 922–928

[38] Qi, C.R., Su, H., Mo, K., 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. (2017) 652–660

[39] Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38**(5) (2019) 1–12

[40] Li, G., Muller, M., Thabet, A., Ghanem, B.: Deepgcns: Can gcns go as deep as cnns? In: Proceedings of the IEEE/CVF international conference on computer vision. (2019) 9267–9276

[41] Xu, M., Ding, R., Zhao, H., Qi, X.: Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 3173–3182

[42] Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 16259–16268

[43] 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 **7**(2) (2021) 187–199

[44] 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. (2019) 3075–3084

[45] Graham, B., Engelcke, M., Van Der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 9224–9232

[46] Hu, W., Zhao, H., Jiang, L., Jia, J., Wong, T.T.: Bidirectional projection network for cross dimension scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 14373–14382

[47] Hu, Z., Zhen, M., Bai, X., Fu, H., Tai, C.l.: Jsenet: Joint semantic segmentation and edge detection network for 3d point clouds. In: European Conference on Computer Vision, Springer (2020) 222–239

[48] Nekrasov, A., Schult, J., Litany, O., Leibe, B., Engelmann, F.: Mix3d: Out-of-context data augmentation for 3d scenes. In: 2021 International Conference on 3D Vision (3DV), IEEE (2021) 116–125

[49] Liu, Z., Qi, X., 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. (2021) 1726–1736

[50] Fabri, A., Pion, S.: Cgal: The computational geometry algorithms library. In: Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems. (2009) 538–539

Appendix for SL3D: Self-supervised-Self-labeled 3D Recognition

A Model Illustrations

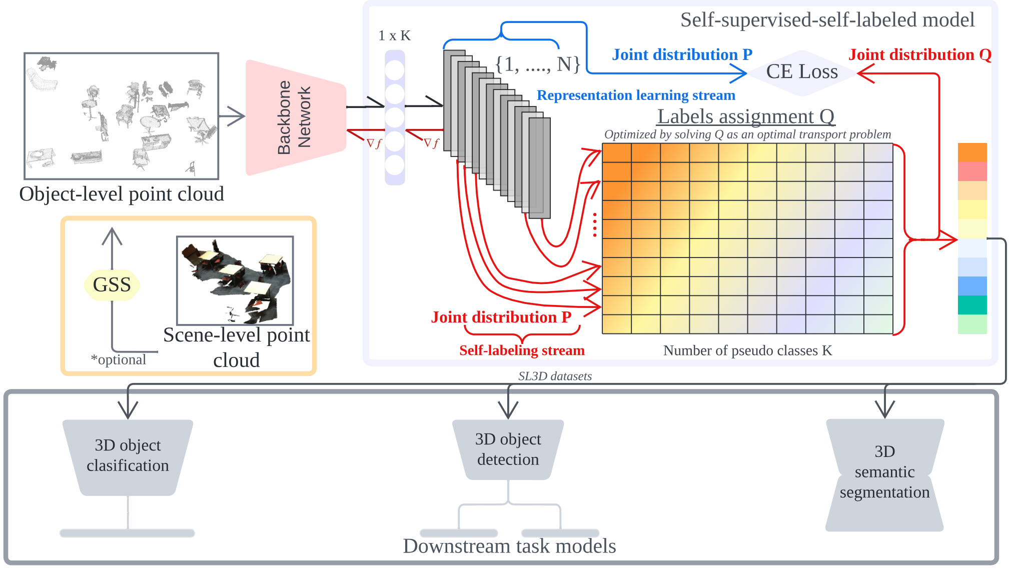


Figure 1: Approach overview: Given object-level point cloud, the model simultaneously clusters and learns feature representation by iteratively cluster features to generate pseudo label and use the labels to guide the feature learning. These pseudo labels can then be used to train several downstream tasks such as 3D object classification, 3D object detection, and 3D semantic segmentation. *For scene-level point cloud data, a geometric selective search algorithm [4] is employed to transform scene data into set of object-level point clouds

B Related Work

In the following, we discuss and highlight existing works for both 3D self-supervised learning and 3D object recognition.

B.1 Self-supervised Learning

A deep learning approach that does not require any supervision signal during learning [16–25, 2, 26, 27]. Instead, it learns visual representations from a given unlabeled dataset. There are several approaches related to 3D self-supervised learning. The contrastive learning approach learns to compare two augmentations of one input sample, and minimizes the embedding distance for features from the same sample and maximizes the embedding distance for those belonging to different samples. Xie *et al.* [28] propose the first transfer learning method for 3D scene understanding by applying contrastive learning on point clouds. Li *et al.* [29] improve semi-supervised segmentation model by utilizing unlabeled data to enhance feature learning. Recently, for self-supervised representation learning from point clouds, different pretext tasks have been exploited to leverage the properties of 3D point cloud. Sauder and Sievers [30] design a model that learns to reconstruct the original point cloud data from randomly rearranged input points for representation learning. Poursaeed *et al.* [31] learn to predict the 3D rotation of objects. Thabet et al. [32] design a recurrent neural network model to predict the next point from a given point sequence generated by a fast space-filling z-order curve. Recently, multi-modality information is also explored in [33] for 3D representation learning by maximizing the agreement in the latent space between point cloud and its 2D correspondence.

The above efforts aim to learn visual representations from given unlabeled data, which can later be used as a pretrained model for further fine-tuning on downstream tasks to alleviate the data requirement. However, in order to solve a 3D recognition task, annotation data is still required in the fine-tuning stage. In contrast with most 3D self-supervised learning models, our framework can
self-label data with clustering and learn visual representations from pseudo labeled data. In this way, not only can the pre-trained model weights be used for further fine-tuning downstream tasks, but we can also utilize the obtained pseudo labels to solve several 3D recognition tasks.

B.2 3D Object Recognition

3D object recognition is one fundamental problem in computer vision that aims to process given 3D visual data to generate high-level understandings. *3D object classification*, *3D object detection*, and *3D semantic segmentation* are three representative tasks.

3D object classification attempts to recognize the class label for a particular object-level point cloud. Early attempts to solve this problem by recognizing multi-view images [34–36] or applying a 3D convolutional network on voxelized 3D data [37, 3]. To avoid generating multi-view images or voxels in the 3D space, PointNet [38] is proposed by Qi *et al.* that directly learns from unordered object-level point cloud data. PointNet++ [6] further enhances the previous model by introducing hierarchical feature learning which improves its robustness towards point density variations. The current state of the art 3D object classification, PointMLP [8], uses a pure residual MLP network and a lightweight geometric affine module. Graph-based learning [39, 40], dynamic neural network [41], and attention architecture [42, 43] are also investigated for point cloud classification.

3D object detection focuses on classifying and locating objects inside a scene-level point cloud by estimating each object's oriented 3D bounding boxes. Upon a PointNet++ backbone for feature extraction, VoteNet [7] designs a voting mechanism to improve bounding box proposal qualities. VoteNet is a representative work in 3D object detection. Recently, unlike [7], RBGNet [13] utilizes a ray-based feature grouping module to improve the 3D bounding box proposal qualities. Weakly-supervised learning on 3D object detection tasks has also been investigated such as [4, 10], which leverage weak annotations as supervision.

3D semantic segmentation focuses on point-level semantic prediction. 3D semantic segmentation methods can be further categorized into point-based approaches [38, 6, 42, 41], which directly extract features from raw input point cloud data and volumetric-based approaches [44–47] , which employ a 3D structured CNN to extract voxelized discrete grid input data. Data augmentation strategies such as Mix3D [48] are also proposed to improve semantic segmentation performance. Recently, weakly supervised learning is also investigated in point cloud segmentation, which learns segmentation from weak labels such as scene category labels [4] and sparse point annotations [49].

These proposed methods all focus on either learning a model from annotated training data or using a self-supervised learning method to improve model feature learning. In contrast, we develop an unsupervised learning method for 3D recognition tasks. Moreover, in our work, we adopt PointNet++ [6] method for both point cloud classification and segmentation and VoteNet [7] method for point cloud object detection.

C Implementation Details

This subsection explains the details of each component in SL3D model, i.e., GSS, Self-supervised-Self-label model, pseudo class annotation, and the downstream task models.

Geometric selective search (GSS). GSS object proposal algorithm is divided into two steps: *Step 1:* detect basic geometry shapes of given scene-level point clouds and *Step 2:* group similar shapes over a region hierarchically based on a similarity score to generate 3D proposals. For detecting geometry shapes, we use the efficient and reliable Computational Geometry Algorithms Library (CGAL) [50]. Specifically, we use the region-growing based algorithm in the shape detection library. We set the parameters of the region-growing based algorithm with a search space of 12 nearest neighbors, the maximum accepted angle between point's normal and plane's normal as 20°, and the minimum region size to 50. For the second step, i.e., the hierarchical agglomerative clustering algorithm, we select size and volume similarity score to generate the 3D bounding box proposals following settings in [4]. Moreover, we apply random jitters for every P_i^S before computing convex hull to improve the proposals recall rate and set the maximum number of proposals to be 1000. Finally, we use the NMS algorithm with an IoU threshold of 0.75 to remove redundant bounding boxes generated by GSS, with all bounding boxes post-processed by the NMS algorithm having a number of point sets smaller than $15k$ points.

Self-supervised-Self-labeled model. SL3D uses two popular backbone networks, i.e., PointNet++ [6] and Point Transformer [42], where the input data with 2048 points of object-level point cloud P_i^O is sampled to 1024 points, and the network output a $K \times 1$ feature vector for each object. We train our model for 600 epochs with batch size 32, a learning rate with an initial value of 0.001, and decayed $10\times$ for every 200 epochs. For optimization, a stochastic gradient descent optimizer is used with a momentum of 0.9 and a weight decay of 0.0001. For the Sinkhorn-Knopp (SK) algorithm, we set λ to be 25 and the number of optimization to be 100. During our model training, we set several pseudo label's numbers $\{18, 40, 50, 100, 200, 400\}$. Our model is computed using 4 NVIDIA GeForce RTX 3090 GPUs, with each unit equipped with 24 Gigabyte memory size.

Pseudo class annotations. To evaluate our model on the standard object recognition benchmark, we need to align the object groups discovered by our model with the number of ground-truth classes. We have to align each of our datasets (depending on the task) by manually group similar pseudo classes into one class label. As shown in Table 4, SL3D requires a large number of pseudo classes to discover all ScanNet semantic classes. There are three reasons behind this. First, it is due to the class imbalance problem in ScanNet dataset, as shown in Fig. 2. In total, ScanNet dataset has $20,096$ object-level point clouds from training ($\sim 78\%$) and validation ($\sim 22\%$) sets where 27.7% of the objects belong to "chair". Second, there is a possibility that the object-level point cloud generated by the GSS algorithm might contain noise, e.g., backgrounds. Therefore, these pseudo classes do not have any corresponding category in the ground truth data. Lastly, SL3D has an equipartition constraint that forces each pseudo class to have an equal number of data. To better understand the influence of pseudo class number towards the actual class distributions, We also shown the comparison of class distribution after grouping process generated from different number of pseudo classes as shown in Fig. 3 and Fig. 4 for both ScanNet and ModelNet datasets respectively.

Table 4: Classes alignment between pseudo labels and ScanNet semantic classes (18 object classes + wall + floor). † denotes SL3D with PointNet++ backbone network while ‡ denotes SL3D with Point Transformer backbone network.

No. pseudo classes	$18^\dagger$	$18^\ddagger$	$50^\dagger$	$50^\ddagger$	$100^\dagger$	$100^\ddagger$	$200^\dagger$	$200^\ddagger$	$400^\dagger$	$400^\ddagger$
No. obtained classes	7	9	9	11	13	14	16	17	18	20

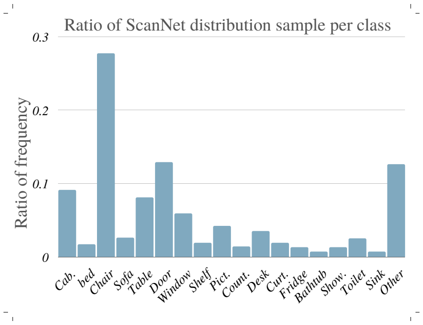


Figure 2: Overview of the class distribution on ScanNet dataset (combined training and validation sets) *without floor and wall classes.

Downstream task models. The downstream task model for both 3D object detection and 3D semantic segmentation can can be any model architecture taking point cloud as inputs. Here, we adopt the classic network for specific tasks.

For the *3D object classification task*, we train PointNet++ [6] classification model with SL3D pseudo labeled data.

For the *3D object detection task*, the VoteNet [7] model is adopted. which takes as inputs raw scene-level point cloud data and is supervised by pseudo labels generated by SL3D. To tailor the detection model for evaluation, an alignment process is conducted to map pseudo labeled data to the corresponding ground-truth categories. We train the model for 180 epochs with a batch size of 32, a learning rate with an initial value of 0.001, and is decayed $10\times$ at epochs $\{80, 120, 160\}$. To optimize the model, an Adam optimization algortihm is used without weight decay.

For the *3D semantic segmentation task*, we choose PointNet++ segmentation model for semantic scene labeling with Multi Scale Grouping (MSG) settings [6]. The model takes as inputs raw scene-level point cloud data and is supervised by pseudo labels from *SL3D*. We train PointNet++ segmentation model for 500 epochs with 32 batch size, a learning rate with an initial value of 0.001, and is decayed $1.5\times$ for every 100 epochs. An Adam optimizer is used with no weigt decay to optimize the model training.

D Ablation Studies

In order to better understand our model, we perform ablation studies on individual components of our model. As GSS [4] is not our contribution, the ablation study of GSS is not included in this ablation study. In the following, we first study the clustering quality of SL3D. Then investigate whether the pretrained SL3D model can learn meaningful representations using supervised finetuning and k-NN evaluation [18]. All ablation studies are performed on the ScanNet dataset.

D.1 Clustering Evaluation

We use ScanNet ground-truth instance object-level point cloud data (training and validation sets) and generate pseudo labels from it using our self-supervised-self-labeled model. To analyze the clustering quality, we use mean purity evaluation measurement For each group, the purity ratio is calculated as the number of objects belonging to the majority class over the total number of data samples. The mean purity ratio is the average of purity ratio over all categories. This is not a perfect metric as it might lead to a trivial solution as purity value of 100% can be achieved by setting the number of pseudo classes to be equals with the total number of data. Considering that the total number of evaluated objects is 20,096, which is much larger than pseudo class number, the mean purity ratio can be used as a proxy metric. We test the model using both PointNet++ and Point Transformer backbone networks with $\{18, 50, 100, 200, 400\}$ number of pseudo-classes as shown in Table 5.

Table 5: Clustering quality (%) on SL3D dataset.

No.	Backbone	**No. SL3D pseudo classes**				
		18	50	100	200	400
1	PointNet++ [6]	40.6	43.5	45.6	47.3	53.9
2	Point Transformer [42]	42.5	47.1	49.9	55.2	61.1

Table 5 shows that our model performs reasonably well under different settings, manifesting the quality of clustering results. The Point Transformer backbone delivers the best result of 61.1% of mean purity ratio with 400 pseudo classes. And the PointNet++ backbone yields a purity ratio of 53.9% with 400 pseudo classes

D.2 Analysis on Pretrained Weights through Finetuning

The purpose of conducting the finetuning evaluation is to assess whether our pre-trained model through SL3D can learn useful representations. First, we trained our self-supervised model with the object-level point cloud data generated by GSS. Second, the pre-trained model weights are used as weight initialization and finetuned on ground-truth object-level point cloud. We use ScanNet object-level point cloud dataset for model training and testing, where $15,733$ objects are allocated for training set and $4,363$ objects for testing set.

As shown in Table 6, we compare two types of model configurations. The first model is the one that uses random initialization, while the second model uses our SL3D pretrained weight as its weight initialization. Moreover, we also show the performance difference when a different type of backbone network is used. We observe that the second model outperforms the first model under all settings (e.g., type of backbone network, number of pseudo classes). This observation shows that the quality of our SL3D's learned representation features helps model to generalize and perform better.

Table 6: Finetuned and k-NN evaluations on SL3D pretrained weights

No.	Weight init.	No. of pseudo	Finetuned eval.	k-NN eval.	
	SL3D pretrained (✓/✗)	classes	Acc (%)	20-NN Top1:Top5	100-NN Top1:Top5
Backbone: PointNet++ [6]					
1	✗	-	82.14	44.76 : 85.40	42.29 : 88.12
2	✓	18	83.76	48.20 : 86.53	45.73 : 88.85
3	✓	50	82.92	48.40 : 86.81	48.20 : 89.61
4	✓	100	83.46	48.88 : 86.95	48:49 : 89.72
5	✓	200	83.95	50.24 : 87.64	49.76 : 90.06
6	✓	400	83.44	52.66 : 88:08	51.48 : 90:31
Backbone: Point Transformer [42]					
7	✗	-	79.64	46.27 : 85.83	45.79 : 88.15
8	✓	18	82.25	59.71 : 93.10	55.74 : 93.87
9	✓	50	81.54	60.98 : 93.17	57.01 : 93.64
10	✓	100	82.35	66.67 : 93.71	63.14 : 94.57
11	✓	200	82.60	69.05 : 95.08	66.67 : 95.49
12	✓	400	80.51	71.34 : 95.39	69.31 : 96.22

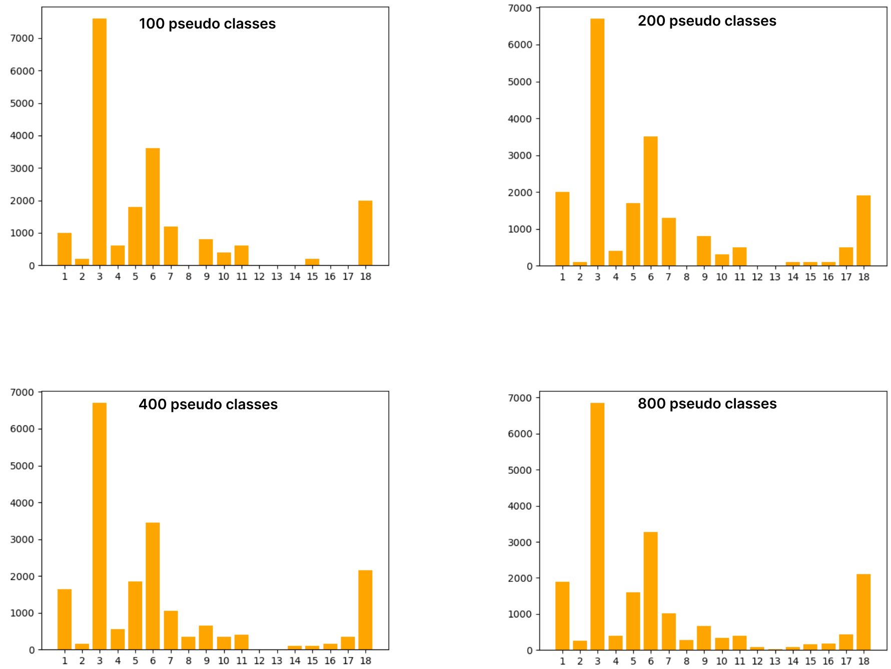
D.3 Analysis on k-NN Evaluation Results

We also experiment on our pretrained model using k-Nearest Neighbour to evaluate the discriminativeness of our features. We divide our experiment into two configurations. The first configuration uses a randomly initialized weight, while the second initializes its weight using SL3D pretrained weights. We evaluate the k-NN model (embedding size of 1024) using ScanNet object-level point cloud data, where $15,733$ objects are allocated for training set and $4,363$ objects for testing set.

The k-NN evaluation result is shown in Table 6, and we show that our SL3D pretrained weights to a great extent help the k-NN model to achieve much higher accuracy and outperform the k-NN model with randomly initialized weight in all settings. We further analyze our pretrained weight by comparing it with a different types of backbone networks. For a randomly initialized weight k-NN model, the 20-NN and 100-NN settings with different backbone networks have accuracy differences (Top 1 : Top 5) of $(1.51 : 0.43)$ and $(3.5 : 0.03)$, respectively, where the model with Point Transformer performs slightly better. With our pretrained weights, the accuracy differences become $(15.87 \pm 3.2 : 6.88 \pm 0.42)$ and $(13.64 \pm 3.63 : 5.04 \pm 0.63)$, respectively. This significant increase shows that SL3D pretrained weights with Point Transformer as its feature extractor effectively helps model learn discriminative features.

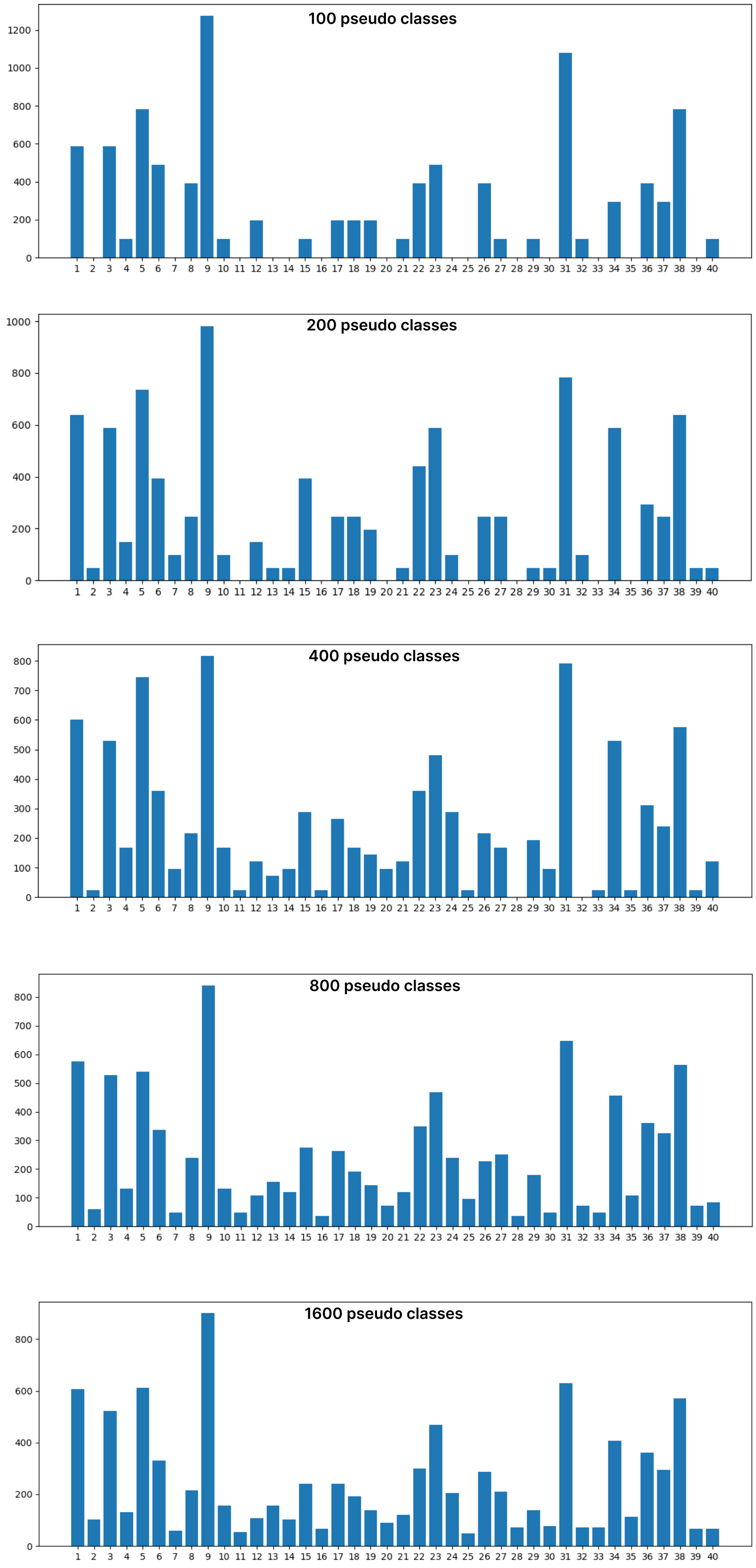
E Limitations and Future Directions

Our research found that SL3D will encounter an issue when the input dataset is imbalanced. This problem occurred as SL3D's self-labeled mechanism has a constraint where the output label classes must have an equally partitioned number of input datasets among the pseudo-classes.



** Vertical axis corresponds to the number of object-level point cloud data while the horizontal axis corresponds to the class index in ScanNet dataset.*

Figure 3: Additional visualization for ScanNet class distribution comparison between different number of pseudo classes.


100 pseudo classes

200 pseudo classes

400 pseudo classes

800 pseudo classes

1600 pseudo classes

* Vertical axis corresponds to the number of object-level point cloud data while the horizontal axis corresponds to the class index in ModelNet40 dataset.

Figure 4: Additional visualization for ModelNet40 class distribution comparison between different number of pseudo classes.