

Learning to Understand Large-Scale 3D Point Clouds



Qingyong Hu
St Hugh's College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity 2022

*This thesis is dedicated to my dearest parents and my girlfriend
for their indispensable, altruistic support.*

Acknowledgements

Looking back on the past three and half years, I had an unforgettable, extraordinary, and meaningful DPhil journey at Oxford. In particular, this experience is even more exceptional due to the outbreak of COVID-19. At the moment I am about to finish my thesis, I would like to show gratitude to the following people who have helped me go through this unusual journey.

First and foremost, I would like to share my deepest appreciation to my supervisor, Professor Andrew Markham and Professor Niki Trigoni, who have always been so encouraging, supportive, and reliable. Thank you for giving me the opportunity to study at Oxford, my dream university, and providing remarkable supervision to lead me from a beginner to a qualified Ph.D. student in this field. I have always known how difficult and challenging that “training from scratch” can be, but your patience, continuous support, careful and encouraging guidance make this possible. I still remember you telling me to dream high and aim higher at the very beginning of my Ph.D., and working on a project which could have a profound impact in the next five to ten years, or even longer. I am even more convinced now that this is absolutely true and visionary. Also, I am also extremely grateful for your open-mindedness and generous support of my research direction, which allows me to deeply and comprehensively explore the challenging and harder research problems, and actively and openly seek cooperation. I could not have imagined having a better advisor as a Ph.D. student.

I would also like to thank all my collaborators, including Dr. Sheikh Khalid who generously provided me with valuable urban-scale point cloud data resources, Prof. Yulan Guo, and Dr. Zhihua Wang who generously share computing resources with me, Prof. Ales Leonardis, Dr. Stefano Rosa, Dr. Linhai Xie, and especially the senior Dr. Bo Yang.

Thanks to all of them for their inspiring discussion, timely and insightful feedback, and spiritual and physical help. In particular, I would like to offer my special thanks to Dr. Bo Yang, as my “closest” collaborator (our workstations are next to each other), he leads me to the exciting land of 3D computer vision. In particular, his enthusiasm and vision for academics, and critical attitude to research questions, have deeply reshaped my taste and pursuit towards meaningful research. I was very fortunate to be able to work together and learn from him at the beginning of my DPhil.

Next, I would like to show appreciation to the labmates of the Cyber-Physical System Group and my lovely friends in the department of computer science and St Hugh’s College. The time with you to go for punting, formal dinners, cooking, drinks, traveling, workout, and playing basketball is always enjoyable and memorable. Thank you for letting me experience the wonderful Oxford life, and thank you for being with me when I am feeling down. This is the best time of my life, and I really dreamed to meet and hang out with all of you again in Oxford in the future.

I am also very grateful to the China Scholarship Council for their financial support, as well as the partial funding support from Amazon Web Services in the Oxford-Singapore Human-Machine Collaboration Programme.

At last, I would like to extend my gratitude to my girlfriend, Runke Huang, for your never-ending encouragement, and for always being there for me. Without your support, I cannot imagine how I am supposed to get through those dark days and sleepless nights. Your support has meant more to me than you could possibly realize.

In the end, I would like to express my sincere thanks to my parents for their meticulous care and wholeheartedly support throughout my life. Your dedication is what keeps me consistently moving forward, with the greatest thanks to you.

Abstract

Giving machines the ability to precisely perceive and understand the 3D visual world is the fundamental step to allow them to interact competently within our physical world. However, the research on large-scale 3D scene understanding and perception is still in its infancy, due to the complex geometrical structure of 3D shapes and limited high-quality data resources. Among various 3D representations, point clouds have attracted increasing attention due to its flexibility, compactness, and the characteristic of being close to raw sensory data. Despite this, the semantic understanding of large-scale 3D point clouds remains a challenge due to its orderless, unstructured, and non-uniform properties. To this end, this thesis makes three core contributions from a high-quality urban-scale dataset, and then fully-supervised semantic understanding to weakly supervised label efficient learning of large-scale 3D point clouds.

The main contributions of this thesis are three-fold. In Chapter 3, we start by building an urban-scale photogrammetric point cloud dataset with nearly three billion richly annotated points. We also identify a number of open and unique challenges faced by urban-scale 3D scene understanding and conduct comprehensive experimental analysis to address these challenges. This work points out the limitations of existing algorithms and provides several insightful conclusions for understanding large-scale 3D point clouds in urban environments.

With high-quality large-scale 3D datasets, we further investigate the research problem of efficient semantic segmentation of large-scale 3D point clouds in Chapter 4. We begin by analyzing the strengths and weaknesses of existing downsampling strategies and find that random sampling is a suitable component for efficient learning on large-scale

point clouds. Additionally, we propose a local feature aggregation module to increase the receptive field hierarchically and preserve important features. We then build an efficient, lightweight neural architecture called RandLA-Net that can directly infer per-point semantics for large-scale point clouds with millions of points that span hundreds of meters.

In Chapter 5, we take a step further to investigate the label-efficient learning of large-scale 3D point clouds. That is, achieving high-quality semantic segmentation with limited annotations. We first examine key issues in weakly supervised learning of 3D point clouds, including different weak supervision schemes and the critical point of weak annotations. Through a pilot study, we find that dense 3D annotations are actually redundant and unnecessary. Motivated by this, we propose a novel weak supervision framework that implicitly augments the total amount of available supervision signals by leveraging the semantic homogeneity between neighboring points. This is achieved using a point neighborhood query to allow the sparse training signals to be back-propagated to a wider context.

All of the algorithms and datasets presented in this thesis have been open-source on GitHub to facilitate future research. The RandLA-Net algorithm was recognized as one of the most influential papers at the CVPR 2020 conference and has been integrated into code libraries such as Open3D and TorchPoints3D. The SensatUrban dataset was used as the platform for two Urban3D challenges, contributing to the advancement of semantic understanding of city-scale point clouds.

Overall, this thesis presents a high-quality dataset and two novel data-driven algorithms, aiming to achieve efficient, scalable, and effective learning-based semantic understanding of large-scale 3D point clouds, eventually improving the real-time 3D perception capacity of intelligent machines in practice.

Contents

1	Introduction	1
1.1	3D Scene Understanding	1
1.2	3D Representation	2
1.3	Scope of the Study and Research Challenges	5
1.4	Contributions & Publications	6
1.5	Thesis Structure	11
2	Literature Review	13
2.1	Background	13
2.2	Semantic Segmentation of 3D Point Clouds with Full Supervision . .	16
2.2.1	Representation Learning for 3D Point Clouds	17
2.2.2	Learning for Large-scale Point Clouds	20
2.3	Semantic Segmentation of 3D Point Clouds with Weak Supervision .	22
2.3.1	Learning with Weak Supervision	22
2.3.1.1	Limited Indirect Annotations	23
2.3.1.2	Limited Point Annotations	24
2.3.2	Learning with Self-Supervision	26
2.4	Datasets for Semantic Segmentation of 3D Point Clouds	27
2.5	Evaluation Metrics	29
2.6	Summary	31
3	Towards Urban-Scale 3D Datasets	35
3.1	Introduction	35
3.2	Dataset Acquisition and Annotation	38
3.2.1	Sequential Aerial Imagery Acquisition	39
3.2.2	Urban-Scale 3D Point Clouds Reconstruction	40
3.2.3	Point-wise Semantic Annotations	41
3.3	Benchmarks	43

3.3.1	Statistics of Train/Val/Test Split	43
3.3.2	Representative Baselines	46
3.3.3	Evaluation Metrics	47
3.3.4	Benchmark Results	47
3.4	Challenges	48
3.4.1	Data Preparation	49
3.4.2	Geometry vs. Appearance	51
3.4.3	The Impact of Skewed Class Distribution	53
3.4.4	Cross-City Generalization	54
3.4.5	Cross-Dataset Generalization	56
3.4.6	Semantic Learning with Fewer Labels	58
3.4.7	Self-supervised Pre-Training on 3D Point Clouds	60
3.5	Discussion	61
3.6	Summary and Outlook	63
4	Learning to Segment Large-Scale 3D Point Clouds	67
4.1	Introduction	67
4.2	Overview	70
4.3	Method	70
4.3.1	The quest for efficient sampling	70
4.3.2	Local Feature Aggregation	74
4.3.2.1	Local Spatial Encoding	74
4.3.2.2	Attentive Pooling	76
4.3.2.3	Dilated Residual Block	77
4.3.3	Network Architecture	78
4.3.4	Implementation	79
4.3.5	Relation to Previous PointNet++	80
4.4	Experiments	82
4.4.1	Efficiency of Random Sampling	82
4.4.2	Efficiency of RandLA-Net	83
4.4.3	Semantic Segmentation on Benchmarks	84
4.4.3.1	Evaluation on Semantic3D	84
4.4.3.2	Evaluation on SemanticKITTI	86
4.4.3.3	Evaluation on Toronto-3D	87
4.4.3.4	Evaluation on NPM3D	89
4.4.3.5	Evaluation on S3DIS	89

4.4.3.6	Evaluation on ScanNet	91
4.4.3.7	Evaluation on DALES	92
4.4.3.8	Evaluation on SensatUrban	92
4.4.4	Ablation Studies	93
4.4.4.1	Ablation of RandLA-Net Framework	94
4.4.4.2	Ablation of LocSE	95
4.4.4.3	Ablation of Dilated Residual Block	95
4.4.5	Analysing Point Neighbours	96
4.4.5.1	Choice of K in KNN	96
4.4.5.2	KNN vs. Spherical Neighbour	97
4.4.5.3	Attention Scores of Neighbouring Points	98
4.4.6	Justification of Design Choices	98
4.4.6.1	Preprocessing	98
4.4.6.2	Nearest-Neighbor Interpolation	100
4.4.6.3	Postprocessing: Voting Scheme	100
4.4.6.4	Sensitivity of RandLA-Net	101
4.4.6.5	Variants of RandLA-Net	101
4.4.6.6	Overfitting issue of RandLA-Net	102
4.5	Discussion	103
4.6	Summary	104
5	Learning to Segment Large-Scale 3D Point Clouds with Fewer Labels	105
5.1	Introduction	105
5.2	Exploring Weak Supervision	108
5.3	Semantic Query Network	111
5.3.1	Overview	111
5.3.2	Point Local Feature Extractor	111
5.3.3	Point Feature Query Network	112
5.3.4	Implementation Details	113
5.3.5	Relations to Previous SuperPoint-based Methods	113
5.4	Experiments	114
5.4.1	Sparse Annotation Tool	114
5.4.1.1	Annotation Pipeline	114
5.4.1.2	The Number of Annotated Points on 7 Datasets	115
5.4.1.3	Annotation Cost	115
5.4.2	Experimental Setup	116

5.4.3	Comparison with SOTA Approaches	117
5.4.3.1	Evaluation on S3DIS	117
5.4.3.2	Evaluation on ScanNet	118
5.4.3.3	Evaluation on Semantic3D	119
5.4.4	Evaluation on Large-Scale 3D Benchmarks	120
5.4.4.1	Evaluation on SensatUrban	120
5.4.4.2	Evaluation on Toronto3D	120
5.4.4.3	Evaluation on DALES	121
5.4.4.4	Evaluation on SemanticKITTI	122
5.4.5	Ablation Study	122
5.4.5.1	Variants of Semantic Queries	123
5.4.5.2	Varying Number of Queried Neighbours	124
5.4.5.3	Varying Annotated Points	125
5.4.5.4	Varying Proportion of Annotated Points	125
5.4.5.5	Varying Backbones of our SQN Framework	126
5.4.5.6	Comparison with Other Feature Propagation Layers	127
5.5	Discussions	128
5.6	Summary	130
6	Conclusion and Future Work	133
6.1	Summary of Key Contributions	133
6.2	Future Work	135
	Bibliography	137

List of Figures

1.1	3D understanding involves a wide range of subtasks, including scene classification, semantic segmentation, object detection, pose estimation, physics based reasoning, saliency prediction, affordance prediction and 3D reconstruction. Sample images adapted from [150].	1
1.2	Illustration of different 3D representations for Stanford bunny. Sample images adapted from [13].	3
1.3	An overview of the thesis structure.	11
2.1	An illustration of point cloud acquisition equipment. The source images are adapted from [18, 9, 12, 19, 3, 20].	13
2.2	An illustration of the unique challenges faced by point clouds. The top left of the figure is adapted from [122], while the bottom left is adapted from [155].	14
2.3	Example applications related to 3D point clouds. Sample images adapted from the internet, the images are credited to [10, 14, 178, 141, 5, 6, 17, 184, 4].	15
2.4	Taxonomy of the field of 3D point cloud semantic segmentation. This figure provides a comprehensive overview of the representative methods and datasets, with a small red flag highlighting the research works presented in this thesis.	16
2.5	Illustration of different 3D representations in semantic segmentation of 3D point clouds. The left of the figure is adapted from [32], while the middle is adapted from [177], and the right is adapted from [231].	17

2.6	We take PointNet as an example to illustrate how existing techniques handle large-scale 3D point clouds. It can be seen that PointNet divides the 3D scene into smaller blocks (left) due to its limitations in processing large-scale 3D point clouds. This leads to inconsistent predictions (right) as the network is unable to comprehend the complete geometry of the point cloud.	20
2.7	A graphical depiction of the progression of data acquisition equipment. It can be seen that advancements in technology have resulted in the collection of increasingly comprehensive data, yet the volume of data has also grown exponentially. Sample images adapted from the internet, the images are credited to [1, 2, 7, 21].	22
2.8	A chronological overview of several representative datasets for 3D point cloud segmentation. The images are adapted from the corresponding datasets [37, 26, 28, 76, 55, 36, 264, 185, 121, 200].	28
2.9	Illustration of IoU metrics for semantic segmentation. Figure from [8].	30
2.10	An overview of the thesis.	32
3.1	This shows an example of urban-scale point clouds in our SensatUrban dataset. It is acquired from the city center of York through UAV photogrammetry. It has a spatial coverage of more than 3 square kilometers and represents a typical urban suburb.	36
3.2	The drones and cameras we used in the urban survey. Sample images adapted from the internet, the images are credited to [15, 16]. . .	39
3.3	An illustration of the survey in a region of Cambridge. A total of 9 flights were carried out together to cover the whole site. Different flight paths of UAVs are represented in lines with different colors. Note that, the drones fly in a grid fashion (<i>i.e.</i> , perpendicular flight) to capture more details of the facades of the urban environment. The circular path is the takeoff and landing pattern.	40
3.4	Visualization of example point cloud tiles in our SensatUrban dataset. Top: the raw point clouds. Bottom: the semantic annotations of corresponding point clouds. Points belonging to different semantic categories are displayed in different colors.	42

3.5	Additional examples of our SensatUrban dataset. Different semantic classes are labeled by different colors. The top two rows are point clouds collected from Birmingham, and the bottom two rows are point clouds acquired from Cambridge.	45
3.6	Statistics of our SensatUrban dataset. The number of points in different semantic categories is reported. Please note that the vertical axis is on the logarithmic scale. Additionally, there are no points annotated as <i>rail</i> in Cambridge.	46
3.7	Qualitative results of PointNet [155], PointNet++ [157], and KPConv [193] on the test set of SensatUrban dataset. The black dashed box highlights the inconsistency predictions with the ground-truth label.	65
4.1	Semantic segmentation results of PointNet++ [157], SPG [110] and our approach on SemanticKITTI [28]. Our RandLA-Net takes only 0.04s to directly process a large point cloud with 10^5 points over $150 \times 130 \times 10$ m in 3D space, which is up to $200 \times$ faster than SPG. Red circles highlight the superior segmentation accuracy of our approach.	68
4.2	In each layer of RandLA-Net, the large-scale point cloud is significantly downsampled, yet is capable of retaining features necessary for accurate segmentation.	70
4.3	The proposed local feature aggregation module. The top panel shows the local spatial encoding block that extracts features, and the attentive pooling mechanism that weights the important neighbouring features, based on the local context and geometry. The bottom panel shows how two of these components are chained together, to increase the receptive field size, within a residual block.	74
4.4	The detailed architecture of our RandLA-Net. $(\tilde{N}, D)$ represents the number of points and feature dimension respectively. FC: Fully Connected layer, LFA: Local Feature Aggregation, RS: Random Sampling, MLP: shared Multi-Layer Perceptron, US: Up-sampling, DP: Dropout.	76
4.5	Illustration of the dilated residual block which significantly increases the receptive field (dotted circle) of each point, colored points represent the aggregated features. L: Local spatial encoding, A: Attentive pooling.	77

4.6	Comparative analysis at module and operation level between our proposed RandLA-Net and PointNet++. A visual representation is provided to highlight the intuitive differences between the two methods. Left: Our RandLA-Net, Right: PointNet++ [157].	80
4.7	Time and memory consumption of different sampling approaches. The dashed lines represent estimated values due to the limited GPU memory. Note that, the curves of GPU memory consumption for RPS, PDS, FPS, and IDIS overlap together, because these sampling methods mainly run on the CPU.	83
4.8	Qualitative results of RandLA-Net on the <i>reduced-8</i> split of Semantic3D. From left to right: full RGB colored point clouds, predicted semantic labels of full point clouds, detailed view of colored point clouds, detailed view of predicted semantic labels. Note that the ground truth of the test set is not publicly available.	85
4.9	Qualitative results of RandLA-Net on the validation set of SemanticKITTI [28]. Red boxes show the failure cases.	88
4.10	Illustration of the 6-fold cross-validation used in the S3DIS dataset. .	90
4.11	Semantic segmentation results of KPConv [193] and our approach on Toronto-3D [185]. Our method clearly performs better on the category of road marking.	91
4.12	The results of our RandLA-Net with different neighbour searching methods. Left: The mIoU scores for different choices of K in KNN. Right: The mIoU scores for different choices of the radius R in spherical neighbours.	97
4.13	Visualization of the learned attention scores in different encoding layers. From top left to bottom right: the learned attention score map in the first (16×16), second (16×64), third (16×128), and the last encoding layers (16×256). For better visualization, we only keep the left part of the attention score map as an 16×16 image. The yellow color represents higher attention scores.	99
5.1	Qualitative results of RandLA-Net (Chapter 4) and our SQN on the S3DIS dataset. Trained with only 0.1% annotations, SQN achieves comparable or even better results than the fully-supervised RandLA-Net. Red bounding boxes highlight the superior segmentation accuracy of our SQN.	106

5.2	Illustration of the sparse annotation tool.	109
5.3	Degradation of three baselines in the <i>Area-5</i> of S3DIS [26] dataset when decreasing proportions of points that are randomly annotated. (Logarithmic scale used in the horizontal axis).	110
5.4	The pipeline of our SQN at the training stage with weak supervision. We only show one query point for simplicity.	111
5.5	Qualitative results achieved by our SQN and the fully-supervised RandLA-Net (Chapter 4) on the <i>Area-5</i> of the S3DIS dataset.	117
5.6	Qualitative results achieved by our SQN on the reduced-8 split of Semantic3D. Note that, the ground truth of the test set is not publicly available.	120
5.7	Qualitative results achieved by our SQN on the validation set (Sequence 08) of SemanticKITTI [28] dataset. The red circle highlights the failure case.	124
5.8	The results of our SQN with different number of querying points on the <i>Area-5</i> of the S3DIS dataset.	125
5.9	Quantitative comparison of RandLA-Net and the proposed SQN on the boundary areas.	129
5.10	Qualitative performance achieved by SQN on the SemanticKITTI dataset when trained on partial point clouds.	129
5.11	Qualitative results achieved by our SQN on the validation set (Sequence 08) of SensatUrban[88] dataset. Best viewed in color.	131

List of Tables

3.1	Detailed specifications of the camera (<i>i.e.</i> , Sensefly SODA 3D camera) used in our survey.	41
3.2	Statistics of the reconstructed 3D point clouds in different cities. The area of the surveyed region and the number of generated points are reported.	41
3.3	Class definitions and ordering of our SensatUrban dataset.	43
3.4	Comparison with existing representative 3D point cloud datasets. ¹ The spatial size (Area/Length) in the dataset, m: meter. Note that, we use distance, instead of area size for outdoor roadway-level datasets. ² The number of classes used for evaluation and the number of sub-classes annotated in brackets. MLS: Mobile Laser Scanning system, TLS: Terrestrial Laser Scanning system, ALS: Aerial Laser Scanning system. Note that, our dataset has total spatial coverage of 7.6 square kilometers, with nearly 3 billion richly annotated points with 13 semantic categories.	44
3.5	Evaluation of selected baselines on our SensatUrban benchmark. We evaluate on 13 predefined semantic categories using the Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%).	48
3.6	Semantic segmentation results achieved by selected baselines [155, 157] with different input preparation steps. Note that, the performance of all baselines is evaluated on the original point clouds, instead of a downsampled point cloud.	49
3.7	Comparison of the grid downsampling and random downsampling at the beginning of data preparation.	51
3.8	Evaluation of semantic segmentation performance of five selected baselines on our SensatUrban dataset with/without the usage of color information.	52

3.9	Evaluation of semantic segmentation performance of PointNet [155] and PointNet++ [157] with different loss functions. To avoid ambiguity and repetition, we use PointNet2 instead of PointNet++ in the Table below.	53
3.10	Cross-city generalization performance of selected baselines on our SensatUrban dataset. All baselines are trained on the <i>training split of Birmingham</i> . The top five records show the testing results on the <i>testing split of Birmingham</i> , while the bottom five rows show the scores on the <i>testing split of Cambridge (cross-city)</i>	54
3.11	Cross-city generalization performance of selected baselines on our dataset. All baselines are trained on the <i>training split of Cambridge</i> . The top five records show the testing results on the <i>testing split of Cambridge</i> , while the bottom five rows show the scores on the <i>testing split of Birmingham (cross-city)</i>	55
3.12	The class mapping from DALES and SensatUrban dataset to the final unified semantic categories.	56
3.13	Statistics of the DALES dataset and SensatUrban dataset after class mapping.	57
3.14	Quantitative cross-dataset generalization results were achieved by the selected baseline approach on the proposed SensatUrban dataset and the DALES dataset.	58
3.15	Quantitative results achieved by PointNet [155] and PointNet++ [157] with different settings (varying number of semantic annotations). [†] means randomly a tiny fraction points of each semantic category, [‡] means randomly a tiny fraction points of the whole points.	58
3.16	Quantitative results achieved by using OcCo [207], Jigsaw [169] and Random (Rand) initialization on the SensatUrban dataset, based on PointNet [155], PCN [245] and DGCNN [213] encoders. Note that, all the initialized weights are obtained by pre-training on the ModelNet40 [224], since these techniques are mainly designed for object-level classification and segmentation.	59

4.1	Comparison of representative sampling methods for processing point clouds. FPS: Farthest Point Sampling; IDIS: Inverse Density Importance Sampling; PDS: Poisson Disk Sampling; RS: Random Sampling; GS: Generator-based Sampling; CRS: Continuous Relaxation based Sampling; PGS: Policy Gradient-based Sampling. Note that, ¹ the complexity denotes the actual computational complexity when sampling M points from a large-scale point cloud P with N points, K denotes the number of nearest neighbours. ² The time consumption is computed for sampling 10% points from 1 million points. We use the same hardware as in Section 3.5, unless specified otherwise.	71
4.2	The run-time decomposition and peak memory of different approaches for semantic segmentation of input point clouds with 12×81920 ($\sim 10^6$) points, which is sampled from the LiDAR point clouds on Sequence 08 (around 80K-120K points per frame) of the SemanticKITTI [28] dataset. In particular, [†] this is achieved by inference half of the input point clouds twice, since PointNet is unable to process the whole inputs in a single pass. [‡] The original implementation of PointNet++ adopts an extremely inefficient KNN during trilinear interpolation. Hence, we also report the result of re-implemented PointNet++* for a fair comparison. Note that, because SPG does not have explicit sampling and neighboring search modules, its run-time decomposition is not reported. We measure the peak memory consumption based on the TensorFlow profiler . Note the peak memory may be different from the actual allocated GPU memory.	84
4.3	Quantitative results of different approaches on Semantic3D (<i>reduced-8</i>) [76]. This test consists of 78,699,329 points. The scores are obtained from the recent publications.	86
4.4	Quantitative results of different approaches on Semantic3D (<i>semantic-8</i>) [76]. This test consists of 2,091,952,018 points. The scores are obtained from the recent publications.	86
4.5	Quantitative results of different approaches on SemanticKITTI [28]. The scores are obtained from the recent publications.	87
4.6	Quantitative results of different approaches on Toronto3D [185]. The scores of the baselines are obtained from [185].	87

4.7	Quantitative results of different approaches on NPM3D [167]. Mean IoU (mIoU, %), and per-class IoU (%) are reported. The scores are obtained from the recent publications.	88
4.8	Quantitative results of different approaches on S3DIS [26] (6-fold cross-validation). Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%) are reported.	89
4.9	Quantitative results of different approaches on ScanNet (online test set).	90
4.10	Quantitative results of different approaches on the DALES dataset. Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%) are reported.	92
4.11	Evaluation of the proposed RandLA-Net [90] on SensatUrban [89] benchmark. We evaluate on 13 predefined semantic categories using the Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%).	93
4.12	The mean IoU scores of all ablated networks based on our full RandLA-Net.	93
4.13	The mean IoU scores of our RandLA-Net with different designs of LocSE.	94
4.14	The mIoU scores of RandLA-Net using different number of aggregation units in each residual block.	96
4.15	Quantitative results of the proposed RandLA-Net on the S3DIS dataset (Area 5) with different preprocessing steps.	99
4.16	Quantitative results of the proposed RandLA-Net on the SemanticKITTI dataset (validation set) with different upsampling strategies.	100
4.17	The mIoU scores (%) of our RandLA-Net with and without the voting scheme.	101
4.18	Quantitative results of the proposed RandLA-Net on the S3DIS dataset (Area 5) with 5 runs. Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%) are reported.	102
4.19	Quantitative results of the proposed RandLA-Net on the SemanticKITTI dataset (validation set) with deeper layers and wider channels.	102
4.20	Training and validation accuracy of our RandLA-Net on three datasets.	103

5.1	A comparison of the total number of points (M: Million) before and after grid sampling for seven public datasets. The grid size and the number of actual annotated points under our 0.1% supervision setting are also reported.	115
5.2	Quantitative results of different methods on the <i>Area-5</i> of S3DIS dataset. Mean IoU (mIoU, %), and per-class IoU (%) scores are reported. Note that, [†] The ratios of these methods represent the ratio of super-point annotations and cannot be directly compared. Bold represents the best result in weak supervision and underlined represents the best in full supervision.	118
5.3	Quantitative results of different approaches on ScanNet (<i>online test set</i>). Mean IoU (mIoU, %), and per-class IoU (%) scores are reported.	118
5.4	Quantitative results of different approaches on Semantic3D (<i>semantic-8</i>) [76]. This test consists of 2,091,952,018 points. The scores are obtained from recent publications. Bold represents the best result in weakly-supervised methods, and underlined represents the best results in fully-supervised methods.	119
5.5	Quantitative results of different approaches on Semantic3D (<i>reduced-8</i>) [76]. This test consists of 78,699,329 points. The scores are obtained from the recent publications. Bold represents the best result in weakly-supervised methods, and underlined represents the best results in fully-supervised methods.	119
5.6	Benchmark results of the baselines on our SensatUrban. Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%) scores are reported. Bold represents the best result in weakly-supervised methods, and underlined represents the best results in fully-supervised methods.	121
5.7	Quantitative results of different approaches on the Toronto3D [185] dataset. The scores of the baselines are obtained from [185]. Bold represents the best result in weakly-supervised methods, and underlined represents the best results in fully-supervised methods.	121
5.8	Quantitative results of different approaches on the DALES dataset. Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%) are reported. Bold represents the best result in weakly-supervised methods, and underlined represents the best results in fully-supervised methods.	122

5.9	Quantitative results of different approaches on SemanticKITTI [28]. The scores are obtained from recent publications. Bold represents the best result in weakly-supervised methods, and underlined rep- resents the best results in fully-supervised methods.	123
5.10	Ablations of different levels of semantic query.	123
5.11	Sensitivity analysis of the proposed SQN on the S3DIS dataset (<i>Area</i> 5) by running 5 times. Overall Accuracy (OA, %), mean IoU (mIoU, %), and per-class IoU (%) are reported. Bold represents the best result.	125
5.12	Quantitative results achieved by our SQN on <i>Area-5</i> of S3DIS under different amounts of labeled points.	126
5.13	Quantitative results achieved by different methods on the region- wise labeled S3DIS dataset.	127
5.14	Quantitative results achieved by our SQN (SPVNAS) on the valida- tion set of the SemanticKITTI dataset under different weak supervi- sion settings.	127
5.15	Ablative experiments on different feature propagation (FP) layers. . .	128

Chapter 1

Introduction

1.1 3D Scene Understanding

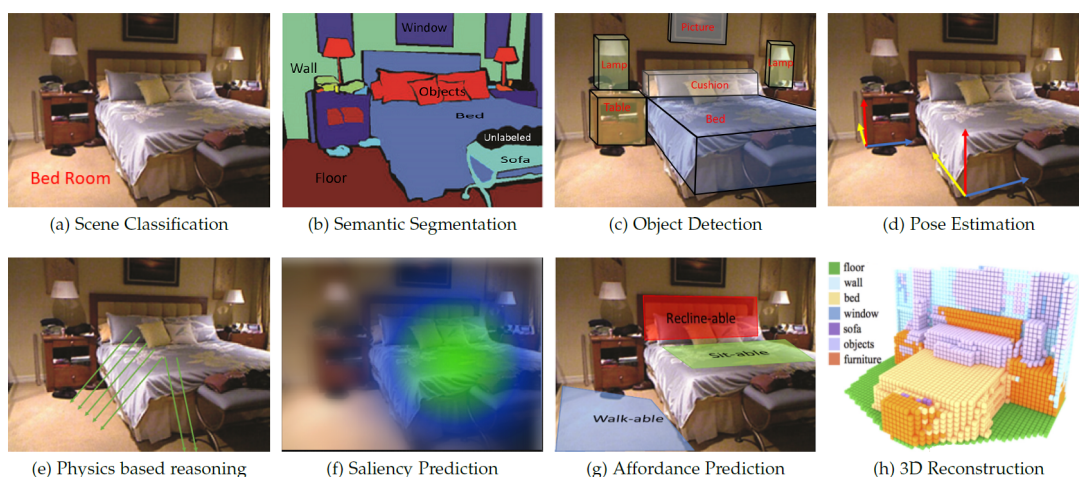


Figure 1.1: 3D understanding involves a wide range of subtasks, including scene classification, semantic segmentation, object detection, pose estimation, physics based reasoning, saliency prediction, affordance prediction and 3D reconstruction. Sample images adapted from [150].

We all live in a 3D world. Although everything humans see is first recorded on our retinas in only two dimensions, we can extrapolate depth and perceive the three-dimensional physical world around us through evolved stereoscopic vision, further recognizing objects, inferring semantic clues, and understanding the structures in 3D space based on our prior knowledge.

In digital worlds, giving machines the ability to perceive and understand the 3D visual world like our humans is also a fundamental and long-standing research question. Due to the limited computational resources and the scarcity of high-quality data resources, early attempts usually start from 2D images with standard

grid structures. Although remarkable progress has been achieved in the past years, the 3D to 2D projection inevitably lost valuable depth information, leading to inferior performance under several intractable cases such as occlusions, viewpoint variations, *etc.*

Recently, 3D scene understanding has been actively explored and widely investigated in several professional areas. On the one hand, various 3D acquisition facilities such as LiDARs, and RGB-D cameras (*e.g.*, Kinect, RealSense, and Apple depth cameras) are becoming increasingly available and affordable. As a result, a number of high-quality 3D datasets are becoming publicly available, including ModelNet [225], ShapeNet [37], ScanNet [55], S3DIS [26], Semantic3D [76], and the KITTI Vision Benchmark Suite [69]. On the other hand, a number of sophisticated techniques within the 3D scene understanding area are being proposed and deployed, enabling various applications including autonomous driving, remote sensing, AR/VR, and the smart city come into our daily life [44, 106, 88].

Formally, 3D scene understanding in machine vision can be defined as: “To analyze a scene by considering the geometric and semantic context of its contents and the intrinsic relationships between them [150].” Overall, 3D scene understanding is a challenging, complex, and systematic research problem, which usually involves a number of sub-tasks including scene classification, object detection, pose estimation, semantic segmentation, 3D reconstruction, saliency detection, physics-based reasoning, and affordance prediction (as illustrated in Figure 1.1). In this thesis, we mainly focus on the fundamental-level subtask and key components in 3D scene understanding, that is, semantic understanding of 3D scenes.

1.2 3D Representation

Representation is a fundamental and key element for machines to understand the 3D digital world [150]. In general, 3D data can be broadly represented by 3D point clouds [155], 3D meshes [80], voxel grids [114], implicit functions [65], primitives [265], and multi-view images [180], as depicted in Figure 1.2. Here, we provide a brief introduction of different representations:

- **Point Cloud Representation.** A “point cloud” is a list of data points in 3D space. Each point is encoded with the spatial coordinates (x , y , z) and possibly with attributes (*e.g.*, remission, reflectance, color information). Point clouds are usually used to represent the surface of a specific 3D geometry. Additionally, point clouds can be acquired by depth sensors and range scanners.

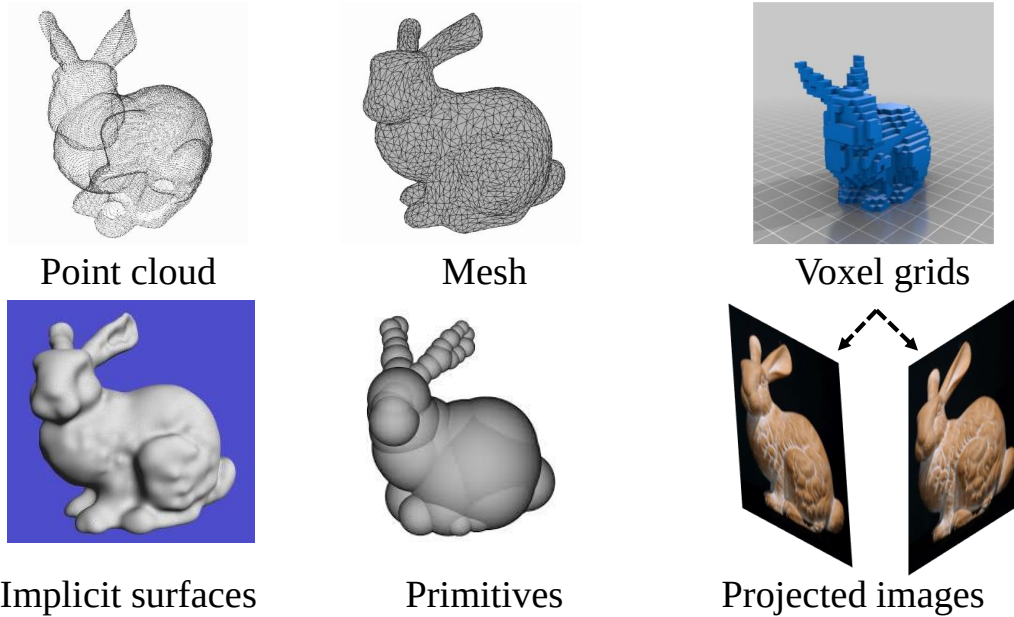


Figure 1.2: Illustration of different 3D representations for Stanford bunny. Sample images adapted from [13].

- **3D Mesh Representation.** The mesh representation is a collection of vertices, edges, and faces that encodes 3D geometry, which has been widely used in 3D computer graphics [214] and geometrical modeling [148]. In particular, polygon mesh means to use polygonal (such as triangle, quadrilateral, or other simple convex polygons) shaped faces to represent the surface of 3D objects. This representation is compact and easy to display and render, but tedious to edit and not friendly for existing deep neural networks with standard operations.
- **Voxel Representation.** Voxel representation is a natural 3D extension of 2D image representation with grid structures, *i.e.*, a voxel represents a regular grid in 3D space. Given a continuous 3D geometric structure (*i.e.*, 3D triangular mesh), voxelization is a process to transform the geometry from a continuous domain into a set of discrete voxels (grids) to best approximate the object. The voxel value is usually binary, where 0 indicates empty and 1 indicates occupied. Due to the structurality and regularity of voxel representation, modern deep neural architectures can be easily applied and extended to this kind of data. However, voxel representation is also voluminous, hence limited to relatively small resolution.
- **Implicit Representation.** Different from the aforementioned explicit 3D representations, implicit 3D surface representation has attracted increasing attention

in recent years for its capacity to represent shape and texture continuously. This representation does not explicitly discretize space and implicitly describes the 3D geometry as the continuous decision boundary of a binary classifier, hence can encode the 3D geometry with extremely high resolution with a fixed memory footprint.

- **Primitives Representation.** Inspired by the nature that humans usually perceive and understand 3D shapes as a collection of simple parts or components [265], this representation describes complex 3D geometry by using a number of basic geometric forms, called primitives (such as cube, cylinder, tube, sphere, cone, *etc.*). Additionally, primitives can also be manipulated (moved, rotated, grouped, hidden, deformed) to better approximate the geometry and improve diversity.
- **Projected Image Representation.** As the name implies, multi-view representation is to encode the 3D geometry as a collection of renderings from multiple different viewpoints [156]. This representation can be easily processed by existing 2D mature neural architectures with standard operations such as CNN. However, the 3D-2D projection inevitably induces information loss, hence limiting the overall performance in some sense.

Among various 3D geometric representations, point clouds have the following desirable properties: 1) Point clouds are natural, straightforward, and compact. They are the closest representation to raw sensory data from LiDAR or RGB-D cameras. 2) They are highly robust to light conditions compared with 2D images, thereby suitable for a wide range of applications. 3) Point clouds do not suffer as much from quantization errors compared to other geometric representations such as voxel grids [156]. They are also flexible and memory-efficient, thereby well-suited for representing details precisely and large-scale 3D scenes. However, deep learning on 3D point clouds is still in its infancy due to the unique challenges faced by point cloud processing, such as being orderless, irregular, and unstructured. Recently, numerous deep learning-based methods have been proposed to address different problems in the area of point cloud analysis, with encouraging results achieved in 3D shape classification and retrieval [155, 157, 122], 3D object detection and tracking [111, 137], and 3D scene segmentation [110, 73].

Although remarkable progress has been achieved in recent years, most existing methods are still limited to extremely small 3D point clouds and cannot be

directly extended to larger point clouds, due to the complex geometrical structure in large-scale 3D scenes and limited high-quality data. This motivates us to build infrastructure, and develop advanced neural architectures to achieve an effective and efficient understanding of unstructured 3D point clouds.

1.3 Scope of the Study and Research Challenges

The scope of this thesis focuses on the advancement of 3D scene understanding, with a specific emphasis on efficient and accurate semantic segmentation of large-scale 3D environments represented in the form of point clouds. This is also the key prerequisite for machines to interact with the 3D environment and perform a range of high-level tasks. Although remarkable progress has been achieved in this research area in recent years, there remain several research challenges that need to be further explored:

- **Can existing techniques achieve semantic understanding of urban-scale 3D scenarios? What are the main challenges?** An essential prerequisite for unleashing the potential of supervised deep learning algorithms in the area of 3D scene understanding is the availability of large-scale and richly annotated datasets. However, publicly available datasets are either in relatively small spatial scales or have limited semantic annotations due to the expensive cost of data acquisition and data annotation, which severely limits the development of fine-grained semantic understanding in the context of 3D point clouds. It remains an open question whether existing approaches can be scaled to urban-scale or even larger 3D scenes. The key challenges of semantic understanding of urban scenes remain unclear.
- **How to understand the semantics of large-scale 3D point clouds efficiently and effectively?** Clearly recognizing individual 3D objects and parsing different semantics within a real-world scene is the prerequisite of 3D scene understanding. In general, a large-scale 3D scene represented by 3D point clouds usually contains dozens of object classes, hundreds of instances, and millions of points. However, most existing methods are still limited to extremely small 3D point clouds and cannot be directly extended to larger point clouds. It remains a challenging problem to develop a flexible, efficient, and effective neural architecture to directly process large-scale 3D point clouds.

- **How to learn the semantics of 3D point clouds with weak (self) supervision?** Existing works usually rely on a large amount of densely annotated semantic labels to provide supervision signals for network training. However, in real-world complex scenarios, it is extremely costly and even infeasible to manually and exactly annotate every point in a whole scene. Therefore, it is desirable to explore the data-efficient learning of 3D point clouds, learning semantics from 3D point clouds with weak (self) supervision (*e.g.*, limited annotations, inexact supervision, pretext-task). Additionally, efficient and effective labeling strategies to significantly reduce the labeling cost are also worth investigating.

Overall, the main goal of my DPhil project is to **semantically understand large-scale 3D point clouds by learning compact and effective representations using deep neural networks with full or limited supervision.**

1.4 Contributions & Publications

In this section, the main contributions of each chapter are summarized as follows. Given the fact that several chapters of this thesis are based on published academic papers, which are developed through collaboration, I will take care to delineate my own role and highlight my specific contributions.

- In Chapter 3, we build an urban-scale photogrammetric point cloud dataset with nearly three billion richly annotated points, termed SensatUrban. Specifically, this dataset consists of large areas from three UK cities, covering about 7.6 km² of the city landscape. In the dataset, each 3D point is labeled as one of 13 semantic classes. We extensively evaluate the performance of state-of-the-art algorithms on our dataset and provide a comprehensive analysis of the results. In addition, we also empirically highlight a number of open challenges in urban-scale scene understanding. The work is published in:

Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, Andrew Markham. "Towards Semantic Segmentation of Urban-Scale 3D Point Clouds: A Dataset, Benchmarks and Challenges". IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021 [88].

Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, Andrew Markham. "SensatUrban: Learning Semantics from Urban-Scale Photogrammetric Point Clouds". International Journal of Computer Vision (IJCV), 2022 [89].

The collaboration for this chapter originated from the Data Study Group held by the Alan Turing Institute, where Sensat Ltd¹ provided the urban-scale 3D point cloud dataset.

My contributions included conceiving the idea of achieving the semantic understanding of urban-scale 3D point clouds and defining the semantic categories of interest for the SensatUrban dataset. I was also responsible for building a comprehensive benchmark, including dividing the dataset into training/validation/testing splits, reporting the statistics and data distributions, and evaluating the performance of six representative baselines (including both projection, voxel, and point-based approaches such as PointNet [155], PointNet++ [157], SparseConv [73], TangentConv [189], KPConv [193], and SPGraph [110]). These algorithms were re-implemented based on their open-source codes.

In addition, I highlighted seven unique and unresolved challenges faced by urban-scale point cloud semantic understanding, including data preparation, geometry *vs.* appearance, skewed class distribution, cross-city generalization, cross-dataset generalization, weakly-supervised learning, and self-supervised pre-training of 3D point clouds. I also provided an in-depth analysis based on extensive experimental results and provided insights to the community for future exploration.

I was also responsible for writing the data acquisition, reconstruction, and annotation process in the manuscript, visualizing the reconstructed point clouds using the Open3D library, and forming the manuscript, which was published in [88, 89].

- In Chapter 4, we introduce an efficient and lightweight neural architecture, termed RandLA-Net, to directly infer per-point semantics for large-scale point clouds. Compared with existing approaches, the proposed RandLA-Net can process 1 million points in a single pass, with up to 200× faster speed than existing approaches on a single NVIDIA RTX2080Ti GPU. Extensive experiments on large-scale point cloud datasets demonstrate the superior performance of our RandLA-Net. The work is published in:

¹<https://www.sensat.co/>, coordinated by Dr. Sheikh Khalid (Chief Scientist)

Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, Andrew Markham. “RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds”. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR Oral)*, 2020 [90].

Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, Andrew Markham. “Learning Semantic Segmentation of Large-Scale Point Clouds with Random Sampling.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021 [91].

For this chapter, I made significant contributions to the design and implementation of the RandLA-Net algorithm for large-scale 3D point cloud semantic segmentation. My responsibilities included the analysis of different downsampling strategies for unstructured 3D point clouds, and decision to employ simple random downsampling in our framework to reduce memory and computational resources. I also designed the lightweight local spatial encoding, dilated residual block unit, built a complementary local feature aggregation module, and developed the RandLA-Net framework with hierarchical downsampling and feature aggregation.

I was also responsible for producing all the experimental results on several large-scale benchmarks, including S3DIS [26], Semantic3D [76], SemanticKITTI [28], Toronto3D [185], ScanNet [55], DALES [200], NPM3D [167], and compared our results with existing approaches such as PointNet [155], PointNet++ [157], KPConv [193], *etc.* I also carried out comprehensive experiments to verify the effectiveness of each component, including visualizing the learned attention scores, efficiency of random sampling, *etc.* Finally, I was also responsible for forming the manuscript published in [90, 91] and creating all figures and tables. My co-authors also made contributions to this project by providing valuable insights and suggestions, the implementation of continuous relaxation-based sampling, policy-gradient-based sampling, and attentive pooling unit.

- In Chapter 5, we take a step further to investigate the label-efficient learning of large-scale 3D point clouds. In particular, we first identified dense 3D annotations are actually redundant and unnecessary, and then propose a new weak supervision framework to implicitly augment the total amount of available supervision signals, by leveraging the semantic homogeneity between neighboring points. Extensive experiments demonstrate the proposed Semantic Query Network (SQN) achieves promising performance on seven

large-scale open datasets under weak supervision schemes, while requiring only 0.1% randomly annotated points for training, greatly reducing annotation cost and effort. The work is published in:

Qingyong Hu, Bo Yang, Guangchi Fang, Yulan Guo, Ales Leonardis, Niki Trigoni, Andrew Markham. "SQN: Weakly-Supervised Semantic Segmentation of Large-Scale 3D Point Clouds". European Conference on Computer Vision (ECCV), 2022 [87].

In this chapter, I made extensive contributions to address the weakly supervised semantic segmentation of large-scale 3D point clouds. My specific contributions included conducting a pilot study to evaluate the redundancy of dense point-wise annotations, providing insight into the critical point of a weak supervision regime. Based on these findings, I developed a novel random sparse annotation scheme that reduces the annotation costs, and created a CloudCompare-based tool to support this approach.

Furthermore, I was responsible for conceiving the idea of leveraging local semantic homogeneity in point neighbors for weakly supervised semantic segmentation of large-scale 3D point clouds, implementing the SQN algorithm using Tensorflow 1.13 and python, and developing a Euclidean distance-based point feature query mechanism. I also produced extensive experimental results on several large-scale benchmarks, including S3DIS [26], Semantic3D [76], SemanticKITTI [28], DALES [200], and Toronto3D [185], demonstrating the effectiveness and efficiency of the SQN algorithm in comparison to existing approaches.

In addition to the technical contributions, I was also responsible for creating the manuscript published in [87], as well as all figures and tables, ensuring the clarity and completeness.

In addition to the above contributions which address the research challenges discussed in Section 1.3, I have also made contributions to the following co-authored publications, which are related to my thesis, but will not be included.

- Yifang Zhang, **Qingyong Hu***, Guoquan Xu, Yanxin Ma, Jianwei Wan, Yulan Guo. *"Not All Points Are Equal: Learning Highly Efficient Point-based Detectors for 3D LiDAR Point Clouds"*. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022 [251].

- Guangchi Fang, **Qingyong Hu**, Hanyun Wang, Yilin Xu, Yulan Guo. [“3DAC: Learning Attribute Compression for Point Clouds”](#). In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022 [63].
- Jia-Xing Zhong, Kaichen Zhou, **Qingyong Hu***, Bing Wang, Niki Trigoni, Andrew Markham [“No Pain, Big Gain: Classify Dynamic Point Cloud Sequences with Static Models by Fitting Feature-level Space-time Surfaces”](#). In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022 [258].
- Meida Chen, **Qingyong Hu***, Thomas Hugues, Andrew Feng, Yu Hou, Kyle McCullough, Lucio Soibelman. [“STPLS3D: A Large-Scale Synthetic and Real Aerial Photogrammetry 3D Point Cloud Dataset”](#). BMVC 2022 [41].
- Taying-Chen, **Qingyong Hu***, Qian Xie, Andrew Markham, Niki Trigoni. [“Meta-Sampler: Almost-Universal yet Task-Oriented Sampling for Point Clouds”](#). ECCV 2022 [50].
- Sheng Ao*, **Qingyong Hu***, Bo Yang, Andrew Markham, and Yulan Guo. [“Spin-Net: Learning a General Surface Descriptor for 3D Point Cloud Registration”](#). In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021 [25].
- Qian Yin*, **Qingyong Hu***, Hao Liu, Feng Zhang, Yingqian Wang, Zaiping Lin, Wei An, Yulan Guo. [“Detecting and Tracking Small and Dense Moving Objects in Satellite Videos: A Benchmark”](#). IEEE Transactions on Geoscience and Remote Sensing (IEEE TGRS), 2021 [244].
- Yulan Guo*, Hanyun Wang*, **Qingyong Hu***, Hao Liu*, Li Liu, and Mohammed Bennamoun. [“Deep Learning for 3D Point Clouds: A Survey”](#). IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2020 [75].
- Kunping Yan*, **Qingyong Hu***, Hanyun Wang, Xiaohong Huang, Li Li, and Song Ji. [“Continuous Mapping Convolution for Large-Scale Point Clouds Semantic Segmentation”](#). IEEE Geoscience and Remote Sensing Letters (IEEE TGRSL), 2021 [235].
- Ruigang Fu, **Qingyong Hu**, Xiaohu Dong, Yulan Guo, Yinghui Gao, Biao Li. [“Axiom-based grad-cam: Towards accurate visualization and explanation of cnns”](#). In British Machine Vision Conference (BMVC), 2020 [64].

- Bo Yang, Jianan Wang, Ronald Clark, **Qingyong Hu**, Sen Wang, Andrew Markham, Niki Trigoni. “*Learning object bounding boxes for 3D instance segmentation on point clouds*”. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019 [238].

1.5 Thesis Structure

The rest of this thesis is organised as follows, and the overall structure of this thesis is depicted in Figure 1.3.

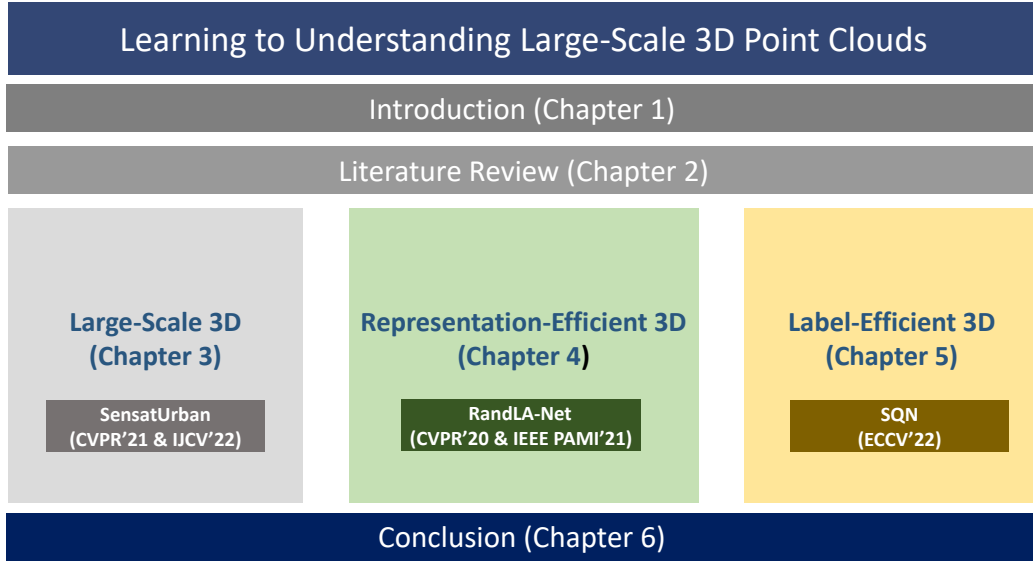


Figure 1.3: An overview of the thesis structure.

- In Chapter 2, we provide a comprehensive literature review of existing 3D semantic understanding methods and related datasets.
- In Chapter 3, we focus on establishing a large-scale 3D dataset, and introducing SensatUrban, an urban-scale photogrammetry point cloud dataset.
- In Chapter 4, we aim to develop an efficient representation for 3D point clouds and present RandLA-Net, a neural architecture for semantic segmentation of large-scale 3D point clouds.
- In Chapter 5, we investigate label-efficient learning algorithms and propose SQN, a weakly supervised semantic segmentation pipeline for large-scale 3D point clouds requiring only 0.1% of semantic annotations.

- In Chapter 6, we conclude the entire thesis and identify a number of future directions.

Chapter 2

Literature Review

2.1 Background

Fundamentally, point clouds are discrete samplings of the scene surface in 3D space, which can be mathematically represented as a collection of three-dimensional point coordinates, and possibly with corresponding attributes. In the last decades, point cloud acquisition equipment has achieved impressive development. In general, point cloud data can be acquired by LiDAR, structured light, stereo vision, or 3D scanner, as shown in Figure 2.1.



Figure 2.1: An illustration of point cloud acquisition equipment. The source images are adapted from [18, 9, 12, 19, 3, 20].

Compared with 2D images, the main advantages of 3D point clouds include:

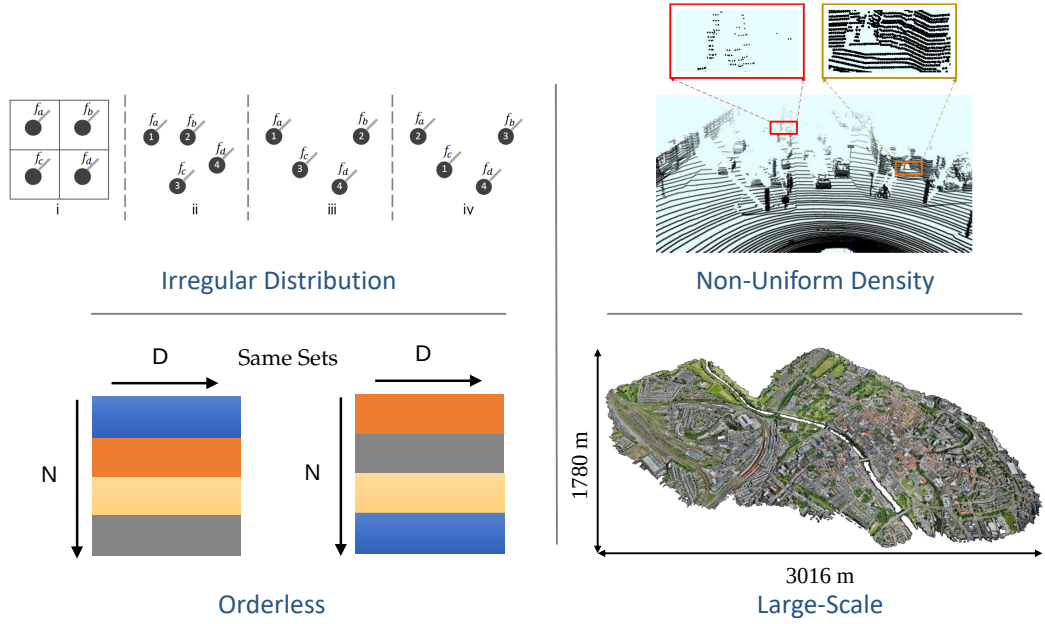


Figure 2.2: An illustration of the unique challenges faced by point clouds. The top left of the figure is adapted from [122], while the bottom left is adapted from [155].

- **3D geometric shape information.** Different from 2D images, which mainly comprise the appearance and texture information, 3D point clouds can provide additional 3D geometrical structure information.
- **No information loss in 3D-2D projection .** Point cloud acquisition is a mapping process from 3D physical space to 3D digital space. Generally, there are only rotation and translation transformations, instead of the projection transformation or distortion in the process of 3D-2D projection, hence insensitive to the viewpoint variations.
- **Robust to light conditions and scale variations.** Both LiDAR and structured light sensors use active lighting, hence the acquisition of point clouds is actually not affected by changes in ambient light. Additionally, considering the point cloud is essentially a discrete sampling of the geometric information of the 3D scene surface, the distance variations only affect the resolution and accuracy of the acquired point clouds, but do not affect the inherent geometric size of the 3D shapes. As such, point clouds are more robust to the scale changes, compared with 2D images.

Despite its benefits, achieving accurate and effective point cloud analysis still faces several unique challenges, as illustrated in Figure 2.2:



Figure 2.3: Example applications related to 3D point clouds. Sample images adapted from the internet, the images are credited to [10, 14, 178, 141, 5, 6, 17, 184, 4].

- **Irregular distribution.** Unlike the regular distribution of image pixels on the 2D plane, the points in a point cloud are usually irregularly distributed in 3D space. In this case, operations such as convolution on images are difficult to directly generalize to 3D point clouds.
- **Non-uniform density.** The distribution of point clouds directly acquired by sensors is usually non-uniform, especially for LiDAR sensors. For example, the point cloud density is high in the area close to the sensor, and is low in the area far away from the sensor.
- **Orderless.** Point cloud data is inherently orderless. Regardless of how the order of the points is shuffled, the 3D geometry (shape) it represents is the same. However, existing neural networks usually assume that there is a fixed order to perform operations, and the results of different orders may not be consistent.
- **Large scale.** The point cloud data collected in real-world 3D scenes are usually large-scale with millions of points, which leads to higher requirements for the calculation and storage efficiency of point cloud processing algorithms.

Recently, deep learning on point clouds has attracted wide attention from the community, with a number of real applications including autonomous driving, AR/VR, interactive entertainment and cultural heritages, as illustrated in Figure 2.3. In this thesis, we mainly focus on the semantic understanding of the 3D point

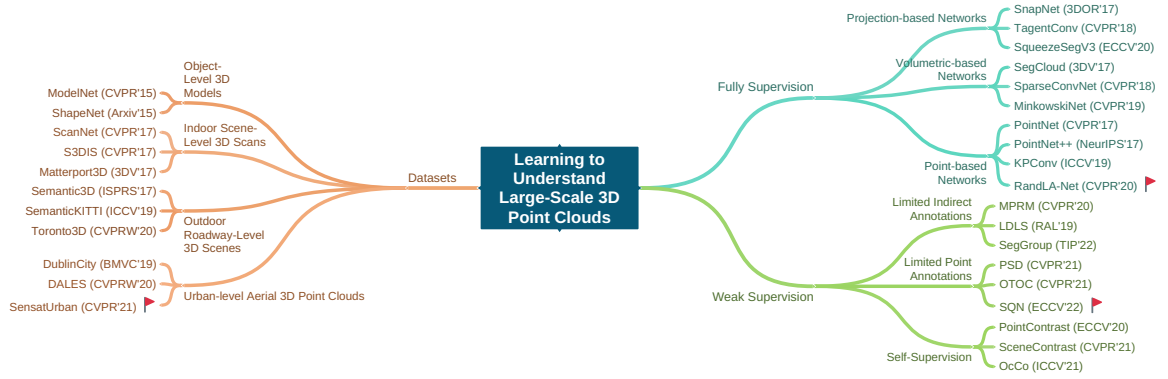


Figure 2.4: Taxonomy of the field of 3D point cloud semantic segmentation. This figure provides a comprehensive overview of the representative methods and datasets, with a small red flag highlighting the research works presented in this thesis.

clouds. Accurate semantic segmentation of large-scale point clouds is a fundamental step of effective 3D scene understanding. This task requires not only the understanding of the global geometric structure, but also the fine-grained local neighborhood details of each point.

Following our research objectives, we have presented a taxonomic classification of the field of 3D point cloud semantic segmentation in Figure 2.4. This taxonomy provides a comprehensive overview of the state and development of the field, facilitating a better understanding of the following literature review. In Sections 2.2 and 2.3, we provide a systematic review of the literature on point cloud semantic segmentation using full and weak supervision, respectively. Furthermore, we also introduce the 3D point cloud datasets (Section 2.4) and evaluation metrics (presented in Section 2.5) used in the field. To conclude this chapter, we summarize the limitations of the existing approaches and provide a brief introduction to the subsequent chapters.

2.2 Semantic Segmentation of 3D Point Clouds with Full Supervision

The task of semantic segmentation for 3D point clouds is to assign labels to individual points and group them based on their semantic meaning. Historically, traditional methods [52, 168] have utilized handcrafted features to extract information from raw point clouds, but recent approaches have shifted towards using

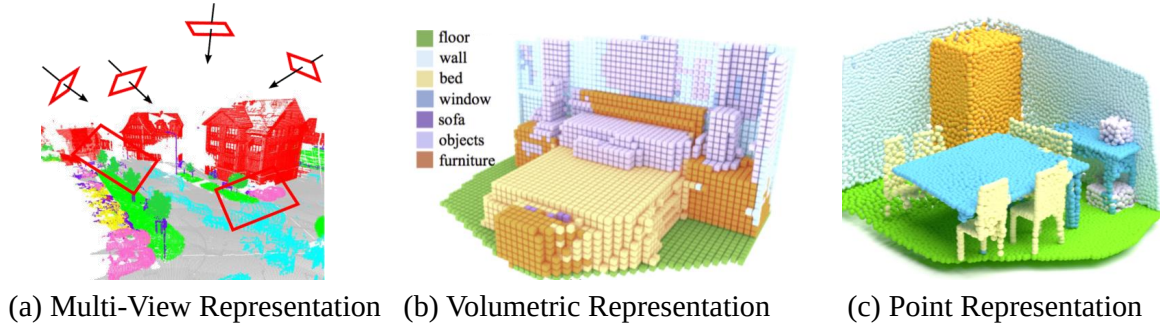


Figure 2.5: Illustration of different 3D representations in semantic segmentation of 3D point clouds. The left of the figure is adapted from [32], while the middle is adapted from [177], and the right is adapted from [231].

deep neural networks, motivated by the great success achieved by deep learning-based techniques in various fields [113]. In this section, our main focus is on the fully supervised scenario where all points have corresponding semantic labels. We will discuss the weakly supervised setting in more detail in Section 2.3.

A major challenge in this task is how to effectively learn meaningful feature representation from irregular, non-uniform, and orderless 3D point clouds. As a result, significant research effort is devoted to developing representation learning techniques that effectively capture the characteristics of these data. Additionally, the large-scale and complex geometrical structure of point clouds necessitates the study of efficient and lightweight neural architectures.

2.2.1 Representation Learning for 3D Point Clouds

To improve feature learning for 3D point clouds, a number of methods have been proposed to learn from irregular raw point clouds or intermediate data representations. These methods can be broadly classified into three categories: projection-based, volumetric-based, and point-based networks. An intuitive illustration can be found in Figure 2.5.

Projection-based networks. To apply the success of 2D CNNs to 3D point clouds, several research works [118, 44, 237, 111] have proposed projecting/flattening 3D point clouds onto 2D images for downstream tasks. However, these approaches can result in a loss of geometric details, particularly when using the commonly-used bird-eye-view projection which can drop points due to occlusion. A representative method is SnapNet [32], as depicted on the left in Figure 2.5. This method generates multiple RGB and depth snapshots of a point cloud using multiple camera positions and employs 2D segmentation networks for pixel-wise labeling of

these snapshots [32]. The scores predicted from the RGB and depth images are then fused through residual correction [27]. However, the challenge of resolving inconsistencies in the prediction results of the same 3D point across different projection maps remains an issue that requires additional investigation.

Alternatively, several other approaches [144, 54, 218, 219, 230] use spherical projection to convert the LiDAR point clouds back to raw range images without dropping points. The Squeezeseg series of techniques [218, 219, 230] are among the most representative methods in this field. These techniques incorporate SqueezeNet [96], an efficient architecture, and a conditional random field or leverage an unsupervised domain adaptation framework to achieve improved segmentation performance on range images. These methods leverage the capability of converting LiDAR point clouds from raw range images obtained through depth sensors, thereby preserving more information than single-view projections. However, as highlighted in [144], the practical implementation of these methods is often faced with challenges such as blurred outputs from convolutional neural networks (CNNs) and quantization errors.

Volumetric-based Networks. Point clouds can also be voxelized into dense 3D grids and then processed with powerful 3D CNNs [46]. Early works [112, 143] usually voxelized point clouds into dense cubic grids, and then leverage the mature 3D CNN architectures to learn the semantics of each point. To achieve fine-grained and globally consistent semantic segmentation, Tchapmi et al. [190] proposed SEG-Cloud. This method involves the use of a deterministic trilinear interpolation to map the coarse voxel predictions generated by a 3D Fully Convolutional Neural Network (3D-FCNN) [135] back to the original point cloud. The inferred per-point labels are then regularized through the use of a Fully Connected Conditional Random Field (FC-CRF) to enforce spatial consistency. Overall, the volumetric representation of 3D point clouds provides inherent preservation of the adjacency relationship and a regular data format, which enables direct application of standard 3D convolutions, resulting in a consistent performance improvement in this field. However, a challenge associated with these techniques is the cubic growth of computation and memory demands with respect to the input resolution, hindering their applicability for processing large-scale point clouds.

To address the issue of cubic growth in computation and memory with increasing input resolution, advanced data structures such as Octree [161] and sparse-tensor [73, 51] have been developed. These structures are beneficial due to the inherent sparsity of the volumetric representation, where only a small percentage

of voxels have non-null values. Graham et al. [73] proposed submanifold sparse convolutional networks that utilize an indexing structure to significantly reduce memory and computational costs. This is achieved by restricting the output of the convolution only related to those occupied voxels. Choy et al. [51] introduced MinkowskiNet, a 4D spatiotemporal convolutional neural network for 3D video perception, which utilizes a generalized sparse convolution to effectively process high-dimensional data. Additionally, different volumetric representations, such as spherical voxels [116] and cylindrical voxels [263], have been proposed to accommodate the unique data distributions of specific point clouds (*e.g.*, LiDAR). Despite the impressive results achieved by these methods in recent years, implementing complex operations can still be a challenge for individual users, especially on GPUs.

Point-based Networks. These methods directly operate on the unstructured point clouds, without relying on any explicit intermediate regular representation. As a pioneering work, Qi et al. [155] introduced PointNet which utilizes simple shared MLPs to extract individual per-point features and symmetrical aggregation functions to ensure permutation invariance. To further capture wider context of each point and learn richer local structures, PointNet++ [157] was proposed. This method groups points in a hierarchical manner and progressively learns from increasingly large local regions. The challenges posed by non-uniformity and varying densities of point clouds were addressed through the introduction of multi-scale and multi-resolution grouping strategies. DGCNN [213] advances the state-of-the-art by presenting a graph-based architecture that models the topological structure through the use of multiple differentiable Edge-Conv operations. Thomas et al. [193] proposed kernel point convolution as a way to learn spatial correlations in unstructured point clouds. Unlike traditional convolution on irregular point clouds, this method performs convolution on predefined regular kernel points.

Point-based methods have gained significant popularity in recent years due to their intuitiveness and simplicity in handling unstructured point clouds. These methods usually use shared MLP as the basic unit to learn point-wise feature, with the goal of learning effective and discriminative 3D representations from irregular point clouds. To this end, various sophisticated neural modules have been proposed based on the point-based framework. In general, these methods can be broadly categorized into four categories: 1) Point-wise MLP methods [155, 157, 99, 61, 255, 254, 240, 228], 2) Point convolution methods [179, 93, 222, 114, 102, 108, 193, 140, 163, 122], 3) RNN-based methods [241, 59, 94, 127], 4) Graph-based

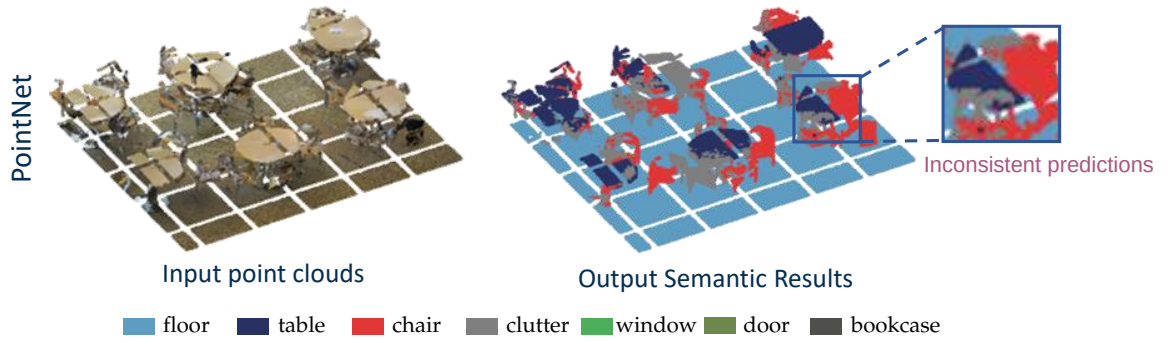


Figure 2.6: We take PointNet as an example to illustrate how existing techniques handle large-scale 3D point clouds. It can be seen that PointNet divides the 3D scene into smaller blocks (left) due to its limitations in processing large-scale 3D point clouds. This leads to inconsistent predictions (right) as the network is unable to comprehend the complete geometry of the point cloud.

methods [110, 109, 208, 257, 128, 119, 213]. A comprehensive literature review of point-based methods can be found in my previous survey paper [75] published in IEEE TPAMI. In this chapter, we will only describe in detail the large-scale point cloud semantic segmentation method that is closely related to our theme.

The simplicity and intuitiveness of point-based methods have made them attractive to the research community. However, most of these approaches suffer from high computational and memory costs, as well as limited scalability when dealing with large-scale point clouds, due to the complex sampling and local feature aggregation operations involved.

2.2.2 Learning for Large-scale Point Clouds

Given the point clouds acquired by depth sensors, another major challenge is to efficiently process large-scale 3D point clouds with millions or even billions of points. This is also crucial for the successful deployment of existing techniques. However, research in this area is still in its infancy and there is currently no established unified framework for addressing this task. As illustrated in Figure 2.6, PointNet cannot directly process the 3D point cloud of the entire room, hence have to divide the room into small blocks and then process them one by one. However, this inevitably causes a loss of overall geometric structure and leads to inconsistent segmentation predictions, as seen in the right top of Figure 2.6, where a table is misclassified as half table and half chair due to the network’s inability to perceive the entire geometry.

Networks for Large-scale Point Clouds. Considering the large-scale and complex geometric structure nature of 3D point clouds, a handful of recent works investigate to direct use of neural networks to process large-scale 3D point clouds. This is often achieved through time-consuming over-segmentation preprocessing or memory-intensive voxelization processing. SPG [110] preprocesses the large point clouds as geometrically-homogenous superpoints, and then builds superpoint graphs to learn per superpoint semantics. However, both the geometric partition and superpoint graph construction are computationally expensive. The recent FCPN [160] and PCT [42] apply both voxel-based and point-based networks to process the massive point clouds. Based on the assumption that points are sampled from locally Euclidean surfaces, TangentConv [189] firstly projects the local surface on the tangent plane and then operates on the projected geometry. Despite being able to process large-scale point clouds, it requires a relatively heavy preprocessing step to calculate the normal.

Sampling Methods for 3D Point Clouds. To reduce the memory and computation cost of processing large-scale 3D point clouds, a handful of research works have started to investigate the point cloud sampling strategies used in neural networks. Existing point sampling approaches [157, 122, 74, 58, 22, 222] can be roughly classified into heuristic and learning-based methods. Farthest point sampling [157, 122] is the most commonly used heuristic sampling strategy in recent works. It iteratively samples the points most distant to the remaining subset from the entire point set. Groh et al. [74] use inverse density importance sub-sampling (IDIS) to preserve points with lower density. Hermosilla et al. [83] utilize Poisson disk sampling (PDS) to achieve a uniform distribution for the sampled points. For learning-based approaches, Yang et al. [240] introduce Gumbel subset sampling, an end-to-end learnable and task-agnostic sampling method, to obtain better performance for downstream tasks. Dovrat et al [58] propose a generator network to directly generate a point set to approximate the original point set. However, most of these methods are either ineffective or computationally heavy for large-scale 3D point clouds.

To summarize, significant progress has been made in point cloud semantic segmentation in recent years, but efficient processing for large-scale 3D point clouds is still an ongoing challenge. First, the question remains on how to design a lightweight and efficient network architecture for direct processing of large-scale point clouds without the need for time-consuming preprocessing. Additionally, another question is on how to design a network architecture that can effectively learn complex

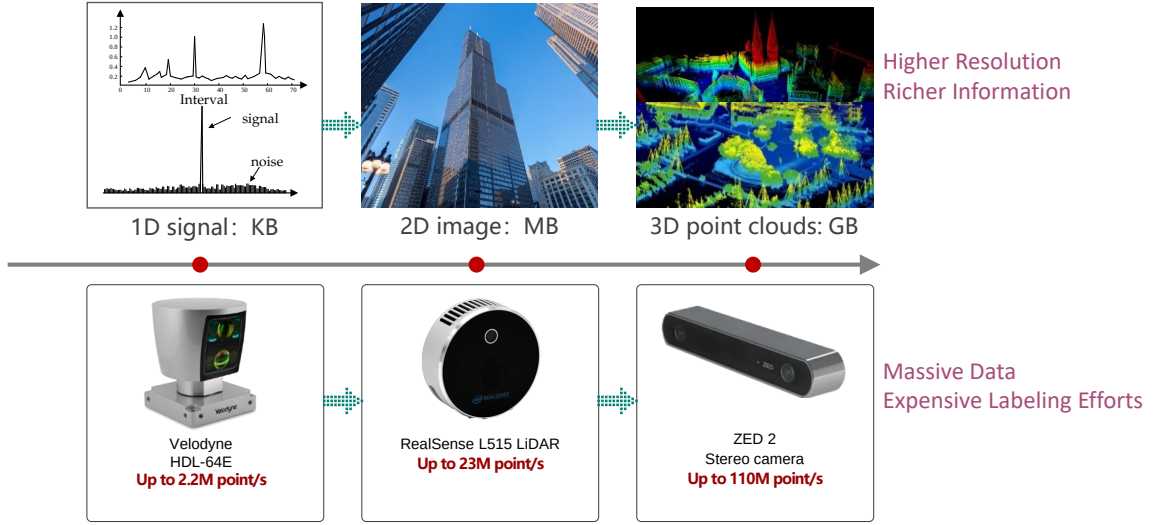


Figure 2.7: A graphical depiction of the progression of data acquisition equipment. It can be seen that advancements in technology have resulted in the collection of increasingly comprehensive data, yet the volume of data has also grown exponentially. Sample images adapted from the internet, the images are credited to [1, 2, 7, 21].

geometric structures in large-scale 3D point clouds. To address these challenges, I presented RandLA-Net in Chapter 4, which is a lightweight and efficient point cloud semantic segmentation algorithm framework.

2.3 Semantic Segmentation of 3D Point Clouds with Weak Supervision

As data acquisition technology improves, it has become possible to collect a wide range of data types, from 1D signals to 2D images to 3D point clouds, with higher resolution and more information, as shown in Figure 2.7. However, this also results in a greater cost for labeling efforts. To address this, research is needed to explore methods of reducing the reliance on labeled data from the algorithm perspective, such as weakly or self-supervised learning for 3D point clouds.

2.3.1 Learning with Weak Supervision

In general, existing weakly supervised point cloud semantic segmentation techniques can be roughly divided into: methods learned from limited indirect annotations and limited point annotations.

2.3.1.1 Limited Indirect Annotations

Instead of using point-level annotations to train segmentation networks, several methods resort to learning from indirect supervision signals, such as 2D image labels, sub-cloud level or scene-level annotations.

Learning with 2D supervision. Considering that point-wise dense annotations are expensive and prohibitive, several approaches have been proposed to learn from relatively cheap 2D image annotations. In [202], a method namely Label Diffusion Lidar Segmentation (LDLS) was proposed to utilize 2D segmentation maps from an aligned camera as supervision signals. A pretrained Mask-RCNN model is first deployed to generate 2D predictions, label diffusion process is further adopted through a graph of connections among 3D points and 2D pixels. Later, Wang et al. [205] propose to leverage sole multi-view 2D segmentation maps, instead of point-wise 3D labels, as the supervision signal to train the 3D semantic segmentation network. This method employs two subnetworks, the Graph-based Pyramid Feature Network (GPFN) and Observability Network (OBSNet), to learn both global and local features, as well as alleviate the occlusion problem caused by projection. In [261], a weakly-supervised CIVA-Net was proposed to learn volumetric segmentation with only image-level class labels. This method first utilizes 3D Grad-CAM to train a classification network and generates pseudo voxel-level labels. A cross-image co-occurrence learning module is further proposed to achieve fine-grained localization maps. Next, an inter-voxel affinity learning module is proposed to generate voxel affinity pairs, which further act as the supervision signal for final semantic segmentation.

To summarize, these methods have demonstrated the feasibility of weakly supervised learning for 3D point clouds using 2D image supervision. However, the performance is relatively limited due to the occlusion and information loss caused by projection.

Learning with 3D indirect supervision. In [215], the authors consider the setting where only sub-cloud level or scene-level labels are available, while the point-level semantic annotations are not. In this case, a method called MPRM was proposed, which first trains a classifier using sub-cloud labels, and then generates point-level pseudo labels using class activation mapping [259]. As a follow-up work, Li et al. [126] extend the MPRM approach to aerial laser scanning point clouds, introducing an overlap region loss to exploit the semantic heterogeneity within a subcloud and an elevation attention unit to improve the quality of the generated pseudo-labels.

Tao et al. [187] present a grouping network for learning semantic and instance segmentation of 3D point clouds, using the seg-level labels generated by over-segmentation pre-processing.

2.3.1.2 Limited Point Annotations

On the other hand, there are a number of methods that have been developed for achieving 3D point cloud semantic segmentation under weak supervision, where only a small fraction of semantically-annotated points or superpoints are available for training. These methods can be broadly classified into four categories: consistency constraint, self-training, label propagation, and active labeling.

Consistency constraint. Consistency training has been demonstrated to be an effective label-efficient learning paradigm. By enforcing prediction consistency between different augmentations/perturbations on the same input data, the unlabeled data could be effectively utilized. Following the consistency training framework, Wu et al. [226] propose a dual adaptive transformations framework, including a point-level local adaptive perturbation module and a regional adaptive deformation module, to enforce the local and structural smoothness constraints on 3D point clouds. Xu and Lee [234] propose a weakly supervised point cloud segmentation method by approximating gradients and using handcrafted spatial and color smoothness constraints. Zhang et al. [249] explicitly added a perturbed branch, and achieve weakly-supervised learning on 3D point clouds by enforcing predictive consistency. In [223], PointMatch was proposed by introducing consistency training to weakly-supervised point cloud semantic segmentation. This method first constructs different views through scene-level data augmentation, and then enforces the consistency of predictions achieved on different views of the same 3D scene. An adaptive pseudo-learning scheme is also proposed to improve the quality of pseudo labels by incorporating the superpoint priors.

Self-Training. Deng et al. [57] propose a superpoint-guided semi-supervised segmentation network based on the self-training framework. This method first combines the geometry-based superpoints and color-based superpoints, which are further used to optimize pseudo labels by leveraging the semantic homogeneity within each superpoint. An edge prediction module is also proposed to constrain the features of edge points. Wang et al. [210] introduce entropy regularization to minimize the entropy of the posterior probability, further decreasing the uncertainty and improving the prediction confidence. An ensemble prediction con-

straint and an online soft pseudo-labeling scheme are also utilized to strengthen the robustness of predictions and leverage additional supervision signals.

Label Propagation. The main idea of this category of methods is to transfer limited supervision signals from labeled points to unlabeled points, hence achieving label-efficient learning on 3D point clouds. In [216], cross-sample and intra-sample feature reallocating modules are introduced to densely propagate supervision signals within and across point cloud samples. A two-stage training strategy is also proposed to rationally take advantage of two modules, while avoiding interference during training. Cheng et al. [48] proposed a weakly-supervised point cloud semantic segmentation method called SSPC-Net. This method first divides the point cloud into superpoints and then utilizes dynamic label propagation and superpoint dropout strategy to generate pseudo-labels for unlabeled superpoints.

Active Labeling. This type of method aims to maximize model performance given limited annotation budgets through active labeling techniques. Shi et al. [172] conducted a systematic analysis of the deployment of active labeling to 3D point cloud segmentation. In particular, the granularity of sample selection and annotation cost metrics are investigated. This paper also identifies that superpoint-level selection is more cost-efficient compared with point-level and instance-level selection schemes. In [92], an active labeling framework namely LiDAL is introduced, which is specially designed for 3D LiDAR point clouds. The key to this method is estimating model uncertainty based on the inconsistency of predictions across frames. In particular, inter-frame divergence and inter-frame entropy are introduced as the uncertainty metrics, and the self-training scheme is further used to improve performance. In [221], a Region-based and Diversity-aware Active Learning (ReDAL) framework is introduced to automatically select informative and diverse small regions as supervision signals. Specifically, softmax entropy, color discontinuity, and structural complexity are used as the criteria for region evaluation.

In addition to the above, several recent works also explore weakly supervised semantic segmentation of 3D point clouds. By observing the strong distribution pattern on the laser beams, Kong et al. [103] proposed LaserMix by leveraging the spatial prior and intertwining laser beams from different LiDAR scans. Specifically, a representation-agnostic mixing scheme is utilized and a mixing-based teacher-student training pipeline is proposed. In [129], a weakly supervised semantic segmentation framework called Less is specially designed for outdoor LiDAR point clouds. This work first identifies that unbalanced class distribution is the major challenge for sparse outdoor LiDAR data through a pilot study, and then

proposes a dedicated framework, including heuristic pre-segmentation, coarsely component proposal annotation, contrastive prototype learning, and multi-scan distillation.

2.3.2 Learning with Self-Supervision

Inspired by the success of self-supervised pre-training representation learning in 2D images [43, 81], a number of very recent studies [132, 252, 191, 97, 45] have started to explore self-supervised learning on 3D point clouds.

Self-supervised Pretraining + Finetuning. These methods usually pretrain the networks on additional 3D source datasets to learn initial per-point representations via self-supervised contrastive losses [227, 132, 207, 252, 191, 169, 97, 45], after which the networks are carefully finetuned on the target datasets with full or partial labels. This noticeably improves the overall performance. Specifically, Saudar and Sievers [169] learn general 3D representation by recovering the correct voxel position of every 3D point after the point cloud is randomly shuffled. Sun et al. propose Canonical Capsules [182] to decompose point clouds into object parts and elements via self-canonicalization and auto-encoding. In [227], a self-supervised representation learning framework termed PointContrast was proposed by taking point cloud registration as the pretext task. Hou et al. [84] further extend this work to fully exploit the spatial context, rather than the vanilla point-level correspondence, and introduce a self-supervised learning framework called Contrastive Scene Contexts. Later, several sophisticated pipelines including Depth-contrast [252], P4contrast [132], 4DContrast [47], and CrossPoint [23] have been proposed. These approaches leverage single/multi-view depth maps and different 3D representations (*i.e.*, point and voxel), pairs of RGBD points as constraints, 4D dynamic sequence information and constraints, and cross-modal instance discrimination for contrastive self-supervised pre-training, respectively.

Self-supervised learning of 3D point clouds has the potential to greatly reduce the cost of annotation by using large amounts of unlabeled data. However, most existing methods are limited to simple objects and failed to be scaled to complex large-scale point clouds. Additionally, there are two main challenges that need to be further addressed: 1) There is a lack of large-scale pre-training datasets for 3D point clouds, similar to ImageNet [105] for 2D images. 2) a general and unified backbone framework is required for 3D point cloud self-supervised learning, such

as ResNet [82] for 2D images. Therefore, future research and development are required in this field.

Although remarkable progress has been achieved in the field of weakly (self-) supervised learning of 3D point clouds in recent years, it remains unclear which weak supervision scheme and what proportion of annotations should use in real applications. Additionally, the strong local semantic homogeneity of point neighbors in large-scale point clouds is not fully exploited, resulting in limited but valuable annotations being underutilized. To address these problems, we systematically explore several critical issues of weakly-supervised learning of 3D point clouds in Chapter 5, including weak supervision scheme, performance variations with annotation ratio, *etc.*, and propose a label-efficient semantic query network to make full use of the semantic similarity between point neighbors.

2.4 Datasets for Semantic Segmentation of 3D Point Clouds

The development of effective deep neural networks is heavily dependent on the availability of high-quality, well-annotated datasets. Existing 3D point cloud datasets can be roughly categorized according to the oriented tasks, for example, datasets used for 3D shape classification [225, 37, 55, 197], datasets used for 3D object detection and tracking [175, 55, 69, 11, 35, 181], and datasets used for 3D point cloud segmentation [28, 170, 167], *etc.* As this thesis primarily focuses on the semantic understanding of large-scale 3D point clouds, we will primarily introduce datasets related to the segmentation task. For a more comprehensive survey of existing point cloud datasets for different tasks, please refer to the survey in [75].

For 3D point cloud segmentation, existing representative datasets can be further categorized into the following four subgroups based on the spatial coverage:

- **Dataset of object-level 3D models.** Early datasets were primarily focused on the recognition of individual objects, and typically consisted of a collection of synthetic 3D CAD models. Representative datasets include the synthetic ModelNet [224], ShapeNet [37], ShapePartNet [243], PartNet [146] and the real-world ScanObjectNN [197]. Such datasets are generally small in spatial size. Additionally, they are also cleaner and more complete than real data, hence commonly used for shape recognition and part segmentation tasks.

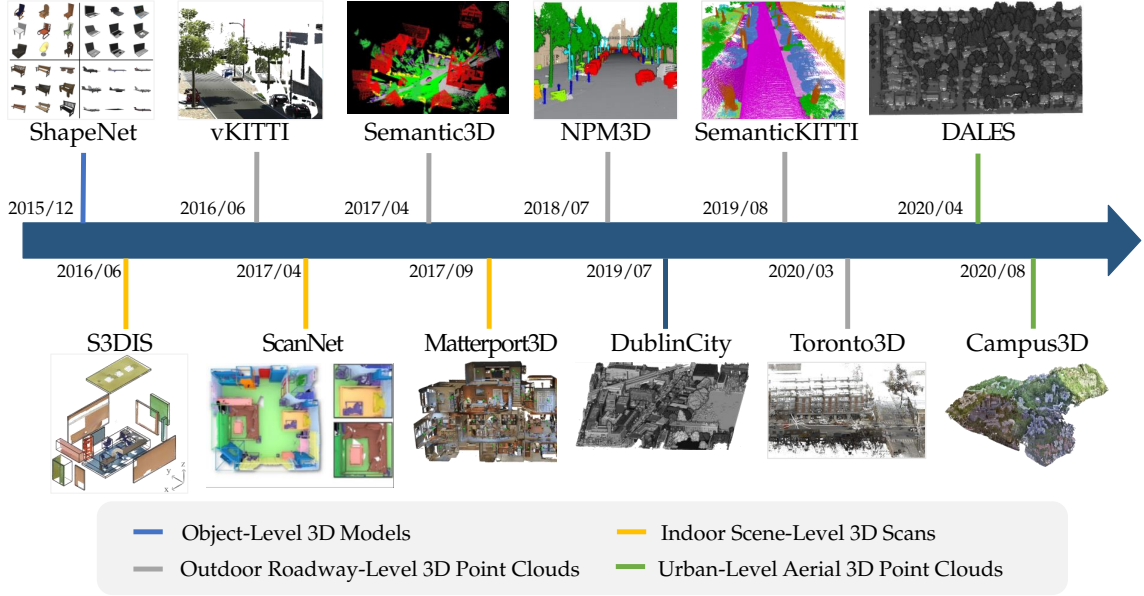


Figure 2.8: A chronological overview of several representative datasets for 3D point cloud segmentation. The images are adapted from the corresponding datasets [37, 26, 28, 76, 55, 36, 264, 185, 121, 200].

However, direct application of the trained model on such data to real environments can be challenging due to the domain gaps between real and synthetic data.

- **Dataset of indoor scene-level 3D scans.** These datasets are usually acquired and further reconstructed by using commodity short-range depth scanners in indoor environments, including NYU3D [173], SUN RGB-D [176], S3DIS [26], SceneNN [260], Matterport3D [36] and ScanNet [55]. Additionally, the SceneNet [79] and SceneNet RGB-D [142] dataset also provide large-scale photo-realistic rendering of indoor synthetic layouts. S3DIS and ScanNet are currently the most commonly used indoor datasets for point cloud semantic segmentation. Notably, ScanNet also offers an online benchmark for evaluating the performance of different semantic segmentation algorithms. Compared with the dataset of object-level 3D models, datasets of 3D indoor point clouds typically contain more semantic categories and cover a larger spatial scale, while presenting a more challenging semantic segmentation task.
- **Dataset of outdoor roadway-level 3D point clouds.** Most of these datasets are driven by the increasing demand of autonomous driving application, and usually collected by using modern laser scanner systems, including static

Terrestrial Laser Scanners (TLS) and Mobile Laser Scanners (MLS). Representative datasets include the early Oakland [149], KITTI [70], Sydney Urban Objects [56] and the recent Semantic3D [76], Paris-Lille-3D [167], Argoverse [38], SemanticKITTI [28], SemanticPOSS [152], Toronto-3D [185], nuScenes [35], A2D2 [72], CSPC-Dataset [194], Lyft dataset¹ and Waymo dataset [181]. Additionally, synthetic datasets [162, 66] composed of realistic simulation of LiDAR point clouds are also included. These datasets are usually larger in spatial scale and more complex in terms of geometry. Additionally, these datasets often contain more noise and outliers, as well as scanning artifacts and incomplete scans.

- **Dataset of urban-level aerial 3D point clouds.** These datasets are usually acquired by professional-grade airborne LiDAR systems, including the recent DublinCity [264], DALES [200] and LASDU [242]. Although these datasets have made significant improvements in spatial scale, a major limitation of these datasets is the lack of color information, which is crucial for fine-grained semantic understanding of 3D scenarios. Recently, the OpenGF [158] dataset has started to investigate ultra-large-scale ground filtering datasets. However, this dataset mainly focuses on the task of ground extraction, instead of fine-grained semantic understanding.

In summary, recent datasets for point cloud semantic segmentation often feature various modalities, increasing spatial scales, more detailed information, and complex geometric structures. These datasets have fueled significant progress in the field of deep learning-based point cloud semantic segmentation. However, there remains a need for high-quality, fine-grained, urban-scale 3D point cloud datasets. To address this gap, we built the SensatUrban dataset, which features urban-scale point clouds with rich semantic annotations and point-wise color attributes, as described in Chapter 3.

2.5 Evaluation Metrics

The performance of semantic segmentation can be evaluated based on various factors such as execution time, memory usage, and accuracy. Here, we will primarily

¹<https://self-driving.lyft.com/level5/data/>



Figure 2.9: Illustration of IoU metrics for semantic segmentation. Figure from [8].

introduce the evaluation metrics used in the subsequent chapters of this thesis, including *Overall Accuracy (OA)*, *mean class Intersection over Union (mIoU)* and *mean class Accuracy (mAcc)*. For more information, please refer to [68].

mean class Intersection over Union (mIoU). Intersection over Union (IoU) is a commonly used metric in object detection to quantify the degree of overlap between two bounding boxes. It is defined as the ratio of the overlapping area to the combined area of prediction and ground truth. In the field of semantic segmentation, the prediction result is not necessarily a rectangle but can have any irregular shape. As a result, IoU in semantic segmentation is more about measuring the accuracy of the predicted point-wise mask in comparison to the ground truth labels, as shown in Figure 2.9. Considering the existence of multiple categories in semantic segmentation, we first calculate the IoU of each class individually. This means that the points belonging to a specific category are considered as the foreground, and the remaining points are considered as the background. The per-class IoU can be calculated by the following equation:

$$IoU_i = \frac{TP_i}{TP_i + FN_i + FP_i} \quad (2.1)$$

where TP_i , FN_i , FP_i is the True Positive, False Negative and False Positive score of class i , respectively. As shown in Figure 2.9, True Positive refers to the area of intersection between the ground truth and the segmentation mask. False Positive refers to the area of the segmentation mask that falls outside of the ground truth, while False Negative refers to the area of the ground truth that the model failed to predict. These scores can be calculated using logical operations such as OR, AND, and their combinations. The IoU values range from 0 to 1, with larger values indicating better results. This metric is particularly useful in semantic segmentation as it takes into account not only the number of correctly classified points but also

the number of false positives and false negatives. Further, the mean class IoU is defined as:

$$mIoU = \frac{1}{N} \sum_{i=1}^N IoU_i \quad (2.2)$$

Overall Accuracy (OA). Similar to the Pascal segmentation convention [62], the accuracy of a class is calculated as:

$$Acc_i = \frac{TP_i}{TP_i + FN_i} \quad (2.3)$$

Further, the overall accuracy of all classes can be defined as:

$$OA = \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N (TP_i + FN_i)} \quad (2.4)$$

mean class Accuracy (mAcc). As its name suggests, mAcc is the average accuracy across all categories, which provides a more comprehensive reflection of the overall performance across different categories.

$$mAcc = \frac{1}{N} \sum_{i=1}^N Acc_i \quad (2.5)$$

2.6 Summary

By reviewing and analyzing relevant research works, we identified several limitations in existing techniques for learning to understand large-scale 3D point clouds. These limitations can be summarized as follows:

- While numerous 3D point cloud datasets have been introduced in recent years, featuring various data modalities, increasing spatial scale, and richer information, there is still a lack of high-quality, richly-annotated urban-scale 3D point cloud datasets. Additionally, the challenges in the semantic understanding of large-scale point clouds remain under-explored. To this end, we proposed an urban-scale photo-realistic photogrammetry point cloud dataset in Chapter 3 and developed a comprehensive benchmark based on this dataset. Through extensive experiments, we comprehensively analyzed the unique challenges faced by semantic understanding of urban-scale point clouds and provided insightful conclusions. To the best of our knowledge, this is the *first*

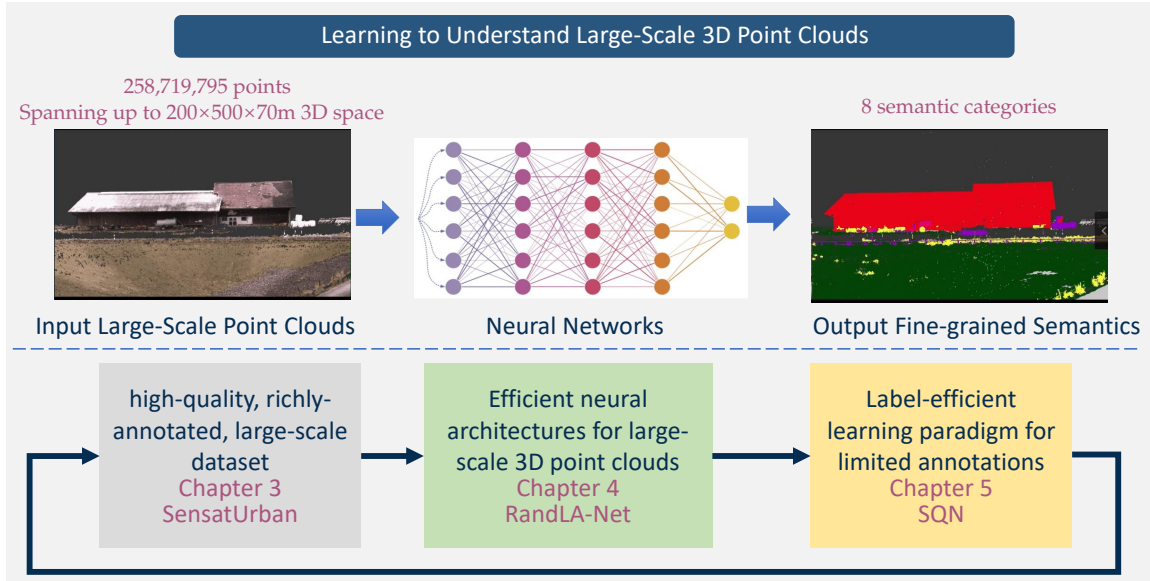


Figure 2.10: An overview of the thesis.

systematic analysis of the research challenges in semantic understanding of large-scale 3D point clouds using deep neural networks.

- Although deep neural networks for 3D point cloud semantic segmentation have made substantial progress recently, most of the existing approaches are either limited to relatively small 3D point clouds (*e.g.*, object-level 3D models or indoor scene-level scenarios), or require time-consuming preprocessing such as over-segmentation to deal with large-scale 3D point clouds. There is still a lack of an efficient semantic segmentation framework for large-scale 3D point clouds. To address this issue, we present an efficient and lightweight framework called RandLA-Net for large-scale 3D point cloud semantic segmentation in Chapter 4. This is achieved by utilizing computationally and memory-efficient random sampling and hierarchical local feature aggregation. In particular, the proposed RandLA-Net achieves competitive segmentation performance, while being up to 200 times faster than other related approaches. .
- Despite significant progress in weakly (self-) supervised learning of 3D point clouds, it is not clear which weak supervision schemes and how much data should be annotated for real-world applications. Additionally, the strong local semantic homogeneity of point neighbors in large-scale point clouds is not fully utilized, leading to the under-utilization of valuable annotations.

To address these challenges, we systematically explore several key issues of weakly-supervised learning of 3D point clouds in Chapter 5, including weak supervision schemes and performance variations with annotation ratios. We propose a label-efficient Semantic Query Network (SQN) that leverages the semantic similarity between point neighbors. This method can achieve close to or even better performance than fully-supervised methods with only 0.1% of random sparse annotations. We also present a random sparse annotation pipeline based on off-the-shelf software, which can save up to 98% of the annotation cost for large-scale 3D point clouds.

Here, we also provide an overview of this thesis, as depicted in Figure 2.10. we begin by building a large-scale 3D point clouds dataset in Chapter 3. Our goal in Chapter 4 is to develop efficient methods for semantic segmentation of these large-scale point clouds. Finally, we delve into the issue of label-efficient learning of large-scale 3D point clouds. This in turn inspires us to build cost-effective large-scale 3D point cloud datasets, since the annotation costs can be significantly reduced.

Chapter 3

Towards Urban-Scale 3D Datasets

3.1 Introduction

Giving machines the ability to semantically interpret 3D scenes is a critical component for accurate 3D perception and scene understanding. This is also a prerequisite for many real-world applications such as object-level robotic grasping [159], scene-level robot navigation [198] and autonomous driving [69], or even large-scale urban 3D modeling, where autonomous machines are required to interact competently within our physical world. The research in this field faces two key challenges: acquiring high-quality datasets and developing effective algorithms. In this chapter, we address the first challenge by establishing high-quality, large-scale 3D point cloud datasets, and the next two chapters will focus on the development of high-performance algorithms.

Recently, an increasingly number of sophisticated neural pipelines have been proposed based on different representations of 3D scenes, including: 1) 3D voxel-based methods such as SegCloud [190], SparseConvNet [73], MinkowskiNet [51], PVCNN [134], Cylinder3D [263] and 2) 2D projection-based approaches such as RangeNet++ [144], SalsaNext [54] and SqueezeSeg [218], PolarNet [250] and 3) recent point-based architectures *e.g.*, PointNet/PointNet++ [155, 157], PointCNN [122], DGCNN [213], KPConv [193], RandLA-Net [90] and PointTransformer [256].

The core of these techniques, however, relies heavily on the wide availability of large-scale and high-quality open datasets. The datasets provide realistic and diverse data resources and act as benchmarks to fairly evaluate and compare the performance of different algorithms. Existing representative 3D data repositories can be generally classified as: 1) object-level 3D models such as ModelNet [224], ShapeNet [37] and ScanObjectNN [197], 2) indoor scene-level 3D scans, *e.g.*, S3DIS [26], ScanNet [55], Matterport3D [36] and SceneNN [260], and 3) outdoor

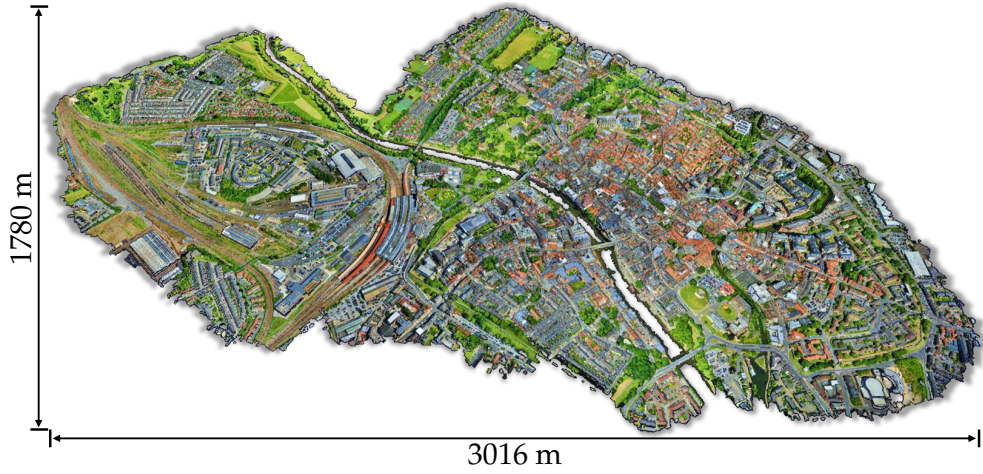


Figure 3.1: This shows an example of urban-scale point clouds in our SensatUrban dataset. It is acquired from the city center of York through UAV photogrammetry. It has a spatial coverage of more than 3 square kilometers and represents a typical urban suburb.

roadway-level 3D point clouds including Semantic3D [76], SemanticKITTI [28], NPM3D [167], and Toronto3D [185].

However, there is no large-scale photo-realistic 3D point cloud dataset available for fine-grained semantic understanding of urban scenarios. Moreover, it remains an open question as to whether existing techniques can be scaled to these urban-scale point clouds. **Firstly**, in contrast to existing datasets for objects, rooms, or streets which are usually less than $200m$ in scale, the urban-scale datasets collected by aerial platforms typically span extremely wide areas, *e.g.*, kilometres. How to efficiently and effectively preprocess massive point sets (*e.g.*, over 10^8) to feed into neural networks is a particular question of interest. **Secondly**, existing photogrammetric mapping techniques allow reconstructing photo-realistic colorized point clouds. Along with the 3D spatial coordinates, is the inclusion of appearance beneficial to semantic understanding and what is the impact if any? **Thirdly**, real-world urban scenarios usually exhibit extreme class imbalance. The majority of points are dominated by categories such as ground and vegetation, while the critical categories such as rail and water only occupy a small proportion of the total number of points. **Fourthly**, and potentially most importantly, what is the generalization performance of existing deep neural networks? Can a trained model be well-generalized to unseen data, particularly from a different region? or even generalized to different dataset? **Lastly**, is it possible to learn semantics with sparser labels? How can we unleash the potential of self-supervised pre-training

and semi-supervised learning on 3D point clouds?

In this chapter, we take a step towards resolving the above issues. In particular, we first build a UAV photogrammetric point cloud dataset called **SensatUrban** for urban-scale 3D semantic understanding. This dataset covers 7.6 km^2 of urban areas in three UK cities *i.e.*, Birmingham, Cambridge, and York (Figure 3.1), along with nearly 3 billion richly annotated 3D points. Each point in the Birmingham and Cambridge set is enriched with one of 13 predefined semantic categories such as *ground, vegetation, car, etc.*, while the points in York remain unlabeled for potential semi-supervised researches. The 3D point clouds are reconstructed from highly overlapped sequential aerial images captured by a professional-grade UAV mapping system. For more detailed data acquisition pipelines, please refer to Section 3.2. Compared with existing 3D datasets, the uniqueness of our SensatUrban lies in two aspects:

- **Urban-scale spatial coverage.** In contrast to existing datasets which mainly focus on objects [224, 37], rooms [260, 26, 55] and roadways [76, 28, 167, 185], the point clouds in our SensatUrban dataset continuously cover several square kilometers of real-world urban areas, opening up new opportunities towards urban-scale applications such as smart cities, and national infrastructure planning and management.
- **Photo-realistic and dense point clouds.** Our dataset is reconstructed from high-resolution aerial images captured by professional calibrated cameras. Unique aerial images from nadir (top-down) and oblique perspectives for the entire landscape of cities are also provided for optimized and high-quality point clouds. Naturally, the geometric patterns, textures, natural colors, point density, and distributions are distinct from existing LiDAR-based datasets.

Based on the proposed SensatUrban dataset, we further highlight several new challenges faced by generalizing existing segmentation algorithms to urban-scale point clouds in Section 3.4. In particular, these challenges include urban-scale data preparation, the usage of color information, learning from extremely imbalanced class distribution, cross-city generalization, and weakly and self-supervised learning from urban-scale point clouds. Note that, this chapter does not aim to fully tackle these challenges, but to unveil them and provide insights to the community for future exploration.

To summarize, the main contributions of this chapter are as follows:

- The creation and introduction of a new, urban-scale photogrammetric point cloud dataset for 3D semantic understanding, with an unprecedented spatial coverage at fine scale and rich semantic annotations.
- A comprehensive benchmark for semantic segmentation of urban-scale point clouds is provided, including extensive experimental results and detailed analysis of several state-of-the-art approaches.
- The identification and discussion of the challenges encountered when generalizing existing neural pipelines to handle extremely large-scale point clouds, along with a comprehensive outlook on future directions for 3D semantic learning.

This work was carried out in collaboration with Sensat¹, where they were responsible for the acquisition, reconstruction of urban-scale photogrammetry point clouds, and we worked together on determining the relevant semantic categories of interest and annotating the urban-scale point clouds through intensive discussion (Section 3.2). My specific contribution includes building the benchmark for urban-scale 3D point cloud semantic segmentation (Section 3.3), conducting extensive experiments, and analyzing the unique challenges faced in semantic segmentation of urban-scale point clouds (Section 3.4), and providing insights and future outlook based on the experimental findings (Section 3.5). Additionally, I also played a key role in composing the manuscript.

This work was first published in [88], and then extended to a journal [89] with more details about the data collection and point cloud reconstruction, additional experimental results and analysis in cross-dataset generalization, and weakly supervised semantic segmentation of urban-scale point clouds. It is also worth mentioning that we have organized two challenges on large-scale point cloud analysis for urban scene understanding, held in ICCV 2021 and ECCV 2022 based on the proposed SensatUrban dataset. This also further boosts the development of deep learning in urban-scale 3D scene understanding. For more details, please refer to <https://urban3dchallenge.github.io/>.

3.2 Dataset Acquisition and Annotation

In this section, we first describe how we collect (Sec. 3.2.1) and reconstruct (Sec. 3.2.2) the urban-scale 3D point clouds using UAV photogrammetry techniques, followed by the detailed procedures to label the dataset over several large urban areas in the UK (Sec. 3.2.3).

¹<https://www.sensat.co/>



Figure 3.2: The drones and cameras we used in the urban survey. Sample images adapted from the internet, the images are credited to [15, 16].

3.2.1 Sequential Aerial Imagery Acquisition

Considering the clear advantages of UAV photogrammetry over similar mapping techniques (such as LiDAR) in terms of cost, data quality, and practicality, we adopt a cost-effective fixed-wing mapping drone, Ebee X², equipped with a cutting-edge Sensefly S.O.D.A. camera, to stably capture high-resolution aerial image sequences, as shown in Figure 3.2. Note that, the camera has the ability to take both oblique and nadir photographs, ensuring that vertical surfaces are captured appropriately. The detailed specification of the camera can be found in Table 3.1.

In order to fully and evenly cover the survey area, all flight paths are pre-planned in a grid fashion and automated by the flight control system (e-Motion). Several factors have been taken into consideration during the data collection workflow: the area covered, the flying permissions, the level of detail required, and the resolution needed, *etc.* In light of the limited battery capacity, multiple individual flights are applied in sequence to capture the whole site (each flight lasts between 40-50 minutes). For illustration, Figure 3.3 shows the paths of the pre-planned multiple flights to cover the selected area in Cambridge city.

These multiple aerial image sequences can then be geo-referenced by Ground Control Points (GCPs) which can be measured by independent professional surveyors with high precision GNSS equipment. Alternatively, the Cambridge data is directly geo-referenced using a highly precise onboard Realtime Kinematic (RTK) GNSS and the final horizontal and vertical RMSEs are ± 50 mm and ± 75 mm, respectively (note they can be improved by introducing GCPs). As a comparison, the expected positioning accuracy of point clouds acquired by airborne LiDAR is

²<https://www.sensefly.com/drone/ebec-x-fixed-wing-drone/>

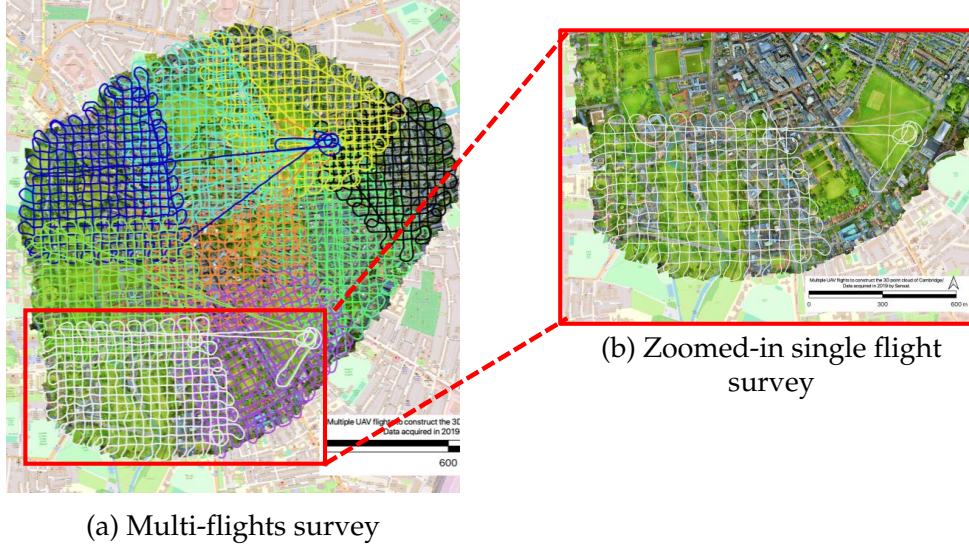


Figure 3.3: An illustration of the survey in a region of Cambridge. A total of 9 flights were carried out together to cover the whole site. Different flight paths of UAVs are represented in lines with different colors. Note that, the drones fly in a grid fashion (*i.e.*, perpendicular flight) to capture more details of the facades of the urban environment. The circular path is the takeoff and landing pattern.

around 5 to 10 cm, depending on the equipment quality, flying configuration, post-processing, *etc.* [253]. The resolution (point density) of our data depends on the number of input images and 3D reconstruction settings. Normally, photogrammetric point clouds are very dense from the process of dense image matching and so need to be subsampled. In our case, all points are subsampled to 2.5 cm, which is denser than most LiDAR data such as DALES [200].

3.2.2 Urban-Scale 3D Point Clouds Reconstruction

Our SensatUrban dataset is reconstructed by using the well-established Structure-from-Motion with Multi-View Stereo (SfM-MVS) techniques[217] on the highly overlapped 2D aerial image sequences. The camera positions and orientation, and the scene geometry are first recovered simultaneously using a highly redundant iterative bundle adjustment, based on matched features extracted from overlapping offset images. The multi-view stereo image matching technique is then applied to reconstruct dense and coloured 3D point clouds.

In this thesis, we use the off-the-shelf software Pix4D³ to generate the 3D point clouds and orthomosaics. The final outputs of the survey include reconstructed

³<https://www.pix4d.com/>

Table 3.1: Detailed specifications of the camera (*i.e.*, Sensefly SODA 3D camera) used in our survey.

		Specification
Sensor size		1 inch
RGB Lens	F/2.8-11, 10.6 mm (35 mm equivalent: 29 mm)	
RGB Resolution		5,472 x 3,648 px (3:2)
Exposure compensation		± 2.0 (1/3 increments)
Shutter		Global Shutter 1/30 – 1/2000s
White balance		Auto, sunny, cloudy, shady
ISO range		125-6400
RGB FOV	Total FOV: 154°, 64° optical, 90° mechanical	
GNSS		RTK/PPK

3D point clouds, 2D orthomosaic images, and 2.5D Digital Surface Model (DSM). In this work, we focus on the 3D point clouds, while the byproduct orthomosaics are only used for visualization purposes. Specifically, we feed all the captured sequential images to Pix4D to generate the 3D point clouds of each region, including the urban area on the periphery of Birmingham, the urban region adjacent to the city centre of Cambridge, and the central area of York. The statistics of the final output point clouds are summarized in Table 3.2.

Table 3.2: Statistics of the reconstructed 3D point clouds in different cities. The area of the surveyed region and the number of generated points are reported.

City	Area (km^2)	Number of Points
Birmingham	1.2	569,147,075
Cambridge	3.2	2,278,514,725
York	3.2	904,155,619
Total	7.6	3,751,817,419

3.2.3 Point-wise Semantic Annotations

To provide fine-grained information of our dataset for subsequent tasks, we further enrich the reconstructed 3D point clouds with point-wise semantic annotations. However, it is non-trivial and particularly important to decide the categories of interest before manual annotation. In this thesis, we specify the semantic categories based on the following three principles: 1) Each annotated category should be of interest to social or commercial purposes, such as asset management [85], automated structural damage assessment [71], and urban planning [86], *etc.* 2) Each



Figure 3.4: Visualization of example point cloud tiles in our SensatUrban dataset. **Top:** the raw point clouds. **Bottom:** the semantic annotations of corresponding point clouds. Points belonging to different semantic categories are displayed in different colors.

category should have a clear and unambiguous semantic meaning. 3) Different categories should have significant variance in terms of geometric structure or appearance.

Based on these three criteria, we first labeled the point cloud as highly detailed 31 categories via off-the-shelf point cloud labeling tools (*i.e.*, CloudCompare), including fine-grained urban elements such as *benches*, *bollards*, *road signs*, *traffic lights*, *etc.* Considering the scarcity of data points in certain categories, we merged some similar categories together and finally identified the below 13 semantic classes for all the 3D points in Birmingham and Cambridge. The detailed definition of the semantic categories are shown in Table 3.3. The points in York remain unlabeled, but made available for possible pre-training in semi-supervised schemes. To ensure the annotation quality, all points are annotated independently by two professional operators in the first round. This is followed by cross-validation in the second round. We also give timely and regular feedback to annotators to address potential issues. All discrepancies in the annotation are carefully addressed, greatly reducing the biases from operators and keeping the annotations consistent and high quality. It takes around 600 working hours to label the entire dataset, and there are no unassigned points discarded in the process. Figure 3.4 shows examples of our annotations. Table 3.4 compares the statistics of our SensatUrban with a number of existing 3D datasets.

Note that, our SensatUrban dataset not only incorporates common categories

Table 3.3: Class definitions and ordering of our SensatUrban dataset.

Class number	Class name	Definition
1	Ground	impervious surfaces, grass, terrain
2	Vegetation	trees, shrubs, hedges, bushes
3	Building	commercial / residential buildings
4	Wall	fence, highway barriers, walls
5	Bridge	road bridges
6	Parking	parking lots
7	Rail	railroad tracks
8	Traffic Road	main streets, highways, drivable areas
9	Street Furniture	benches, poles, lights
10	Car	cars, trucks, jeeps, SUVs, HGVs
11	Footpath	walkway, alley
12	Bike	bikes / bicyclists
13	Water	rivers / water canals

such as *ground*, *building*, and *vegetation*, but also involves several new categories that were not included in the previous urban-scale point cloud datasets [200, 264], such as *rail*, *bridge*, and *water*. In particular, these categories are derived by discussing with the industry professionals, and are particularly important for urban planning and infrastructure mapping.

The SensatUrban dataset has been made publicly available⁴, all point clouds and the point-wise ground-truth label of the training set are provided for network training, and an online hidden test set⁵ is used to evaluate the final segmentation performance. To prevent overfitting on the test set, the maximum number of submissions is also limited.

3.3 Benchmarks

3.3.1 Statistics of Train/Val/Test Split

Based on the proposed SensatUrban dataset, we further set up a benchmark to evaluate the performance of existing state-of-the-art segmentation methods. Notably, we first follow DALES [200] to divide the urban-scale point clouds into similarly sized small tiles (without overlap), so that existing methods can be trained and tested on modern GPUs. Specifically, the urban point clouds collected in Birmingham have been split into 14 tiles, and the Cambridge point clouds are similarly

⁴<http://point-cloud-analysis.cs.ox.ac.uk>

⁵<https://competitions.codalab.org/competitions/31519>

Table 3.4: Comparison with existing representative 3D point cloud datasets. ¹The spatial size (Area/Length) in the dataset, m: meter. Note that, we use distance, instead of area size for outdoor roadway-level datasets. ²The number of classes used for evaluation and the number of sub-classes annotated in brackets. MLS: Mobile Laser Scanning system, TLS: Terrestrial Laser Scanning system, ALS: Aerial Laser Scanning system. Note that, our dataset has total spatial coverage of 7.6 square kilometers, with nearly 3 billion richly annotated points with 13 semantic categories.

	#Name and Reference	#Year	#Spatial size ¹	#Classes ²	#Points	#RGB	#Sensors
Object-level	ModelNet [224]	2015	-	40	-	No	Synthetic
	ShapeNet [37]	2015	-	55	-	No	Synthetic
	PartNet [146]	2019	-	24	-	No	Synthetic
	ScanObjectNN [146]	2019	-	15	-	Yes	RGB-D
Indoor	S3DIS [26]	2017	$6 \times 10^3 m^2$	13 (13)	273M	Yes	Matterport
Scene-level	ScanNet [55]	2017	$1.13 \times 10^5 m^2$	20 (20)	242M	Yes	RGB-D
	Paris-rue-Madame [170]	2014	$0.16 \times 10^3 m$	17	20M	No	MLS
Outdoor Roadway-level	IQmulus [199]	2015	$10 \times 10^3 m$	8 (22)	300M	No	MLS
	Semantic3D [76]	2017	-	8 (9)	4000M	Yes	TLS
	Paris-Lille-3D [167]	2018	$1.94 \times 10^3 m$	9 (50)	143M	No	MLS
	SemanticKITTI [28]	2019	$39.2 \times 10^3 m$	25 (28)	4549M	No	MLS
	Toronto-3D [185]	2020	$1 \times 10^3 m$	8 (9)	78.3M	Yes	MLS
	ISPRS [164]	2012	-	9	1.2M	No	ALS
Urban-level	DublinCity [264]	2019	$2 \times 10^6 m^2$	13	260M	No	ALS
	DALES [200]	2020	$10 \times 10^6 m^2$	8 (9)	505M	No	ALS
	LASDU [242]	2020	$1.02 \times 10^6 m^2$	5	3.12M	No	ALS
	Campus3D [121]	2020	$1.58 \times 10^6 m^2$	24	937.1M	Yes	UAV Photogrammetry
	SensatUrban (Ours)	2020	$7.64 \times 10^6 m^2$	13 (31)	2847M	Yes	UAV Photogrammetry



Figure 3.5: Additional examples of our SensatUrban dataset. Different semantic classes are labeled by different colors. The top two rows are point clouds collected from Birmingham, and the bottom two rows are point clouds acquired from Cambridge.

divided into 29 tiles in total. Note that, each tile is approximately 400×400 square meters. We also report the detailed statistics of the training/validation/testing subsets in both Birmingham and Cambridge in Figure 3.6. It can be seen that the number of points belongs to different semantic categories varies greatly. For example, the dominant three semantic categories, *i.e.*, *ground* / *building* / *vegetation*, together account for more than 50% of the total points. However, the minor yet important two categories (*e.g.*, *bike* / *rail*) only account for 0.025% of the total points. This clearly shows that the class distribution of our SensatUrban dataset is extremely imbalanced, which is in correspondence with the long-tailed distribution of real data. As described in Sec. 3.3, the imbalanced distribution nature of our dataset also poses great challenges in generalizing the existing segmentation

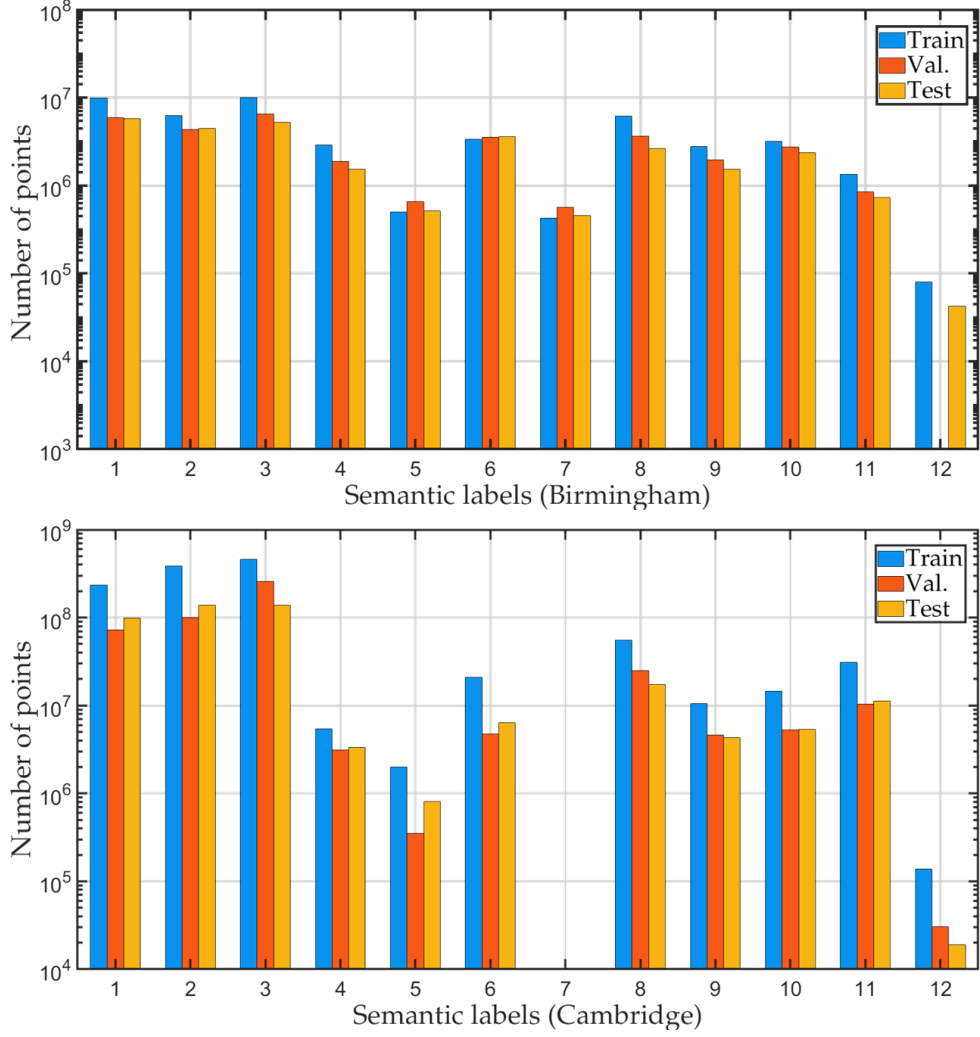


Figure 3.6: Statistics of our SensatUrban dataset. The number of points in different semantic categories is reported. Please note that the vertical axis is on the logarithmic scale. Additionally, there are no points annotated as *rail* in Cambridge.

approaches.

3.3.2 Representative Baselines

To comprehensively evaluate the performance of existing point cloud segmentation pipelines on urban-scale point clouds, we carefully select 6 representative approaches, which cover the three mainstream paradigms as discussed in Section 2.2, as solid baselines in our SensatUrban benchmark. A short summary of these baselines are as follows:

- SparseConvNet [73]. A strong baseline that introduces the submanifold sparse

convolutional networks for efficient semantic segmentation of 3D point clouds. This method and its follow-up works [51, 78] lead the ScanNet benchmark.

- TangentConv [189]. This method introduces tangent convolution, which operates directly on surface geometry, for large-scale point clouds processing. In particular, the 3D point clouds are first projected as tangent images, followed by 2D convolutional networks.
- PointNet [155]. This is the pioneering work for directly operating on orderless point clouds by using shared MLPs and symmetrical max-pooling aggregation.
- PointNet++ [157]. This is the follow-up work of PointNet. It introduced multi-scale/resolution grouping to extract local geometrical patterns, and farthest point sampling to reduce memory and computational cost.
- KPConv [193]. This approach presents a powerful kernel point convolution to learn spatially correlation from unstructured 3D point clouds. A set of rigid or deformable kernel points are placed to learn varying local geometries. It has achieved state-of-the-art performance on the aerial DALES dataset [200].
- SPGraph [110]. This is one of the first learning-based frameworks that capable of processing large-scale point clouds with millions of points. The pipeline composed of geometrically homogeneous partition, followed by superpoint graph construction and contextual segmentation. It is one of the top-performing approaches on the Semantic3D dataset.

3.3.3 Evaluation Metrics

Similar to most of the existing 3D point cloud benchmarks [76, 28, 26], *Overall Accuracy* (OA), *mean class Accuracy* (mAcc), and *mean Intersection-over-Union* (mIoU) are adopted as the primary evaluation criteria of our SensatUrban benchmark. The detailed formulation of these metrics could be found in Section 2.5.

3.3.4 Benchmark Results

We then evaluate the performance of the aforementioned baselines on our urban-scale SensatUrban dataset. The quantitative results including the per-class IoU scores are reported in Table 3.5. We strictly followed the publicly available implementation of each baseline as described in their respective original manuscripts.

Table 3.5: Evaluation of selected baselines on our SensatUrban benchmark. We evaluate on 13 predefined semantic categories using the Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%).

	OA(%)	mAcc(%)	mIoU(%)	ground	veg.	building	wall	bridge	parking	rail	traffic.	street.	car	footpath	bike	water
PointNet [155]	80.78	30.32	23.71	67.96	89.52	80.05	0.00	0.00	3.95	0.00	31.55	0.00	35.14	0.00	0.00	0.00
PointNet++ [157]	84.30	39.97	32.92	72.46	94.24	84.77	2.72	2.09	25.79	0.00	31.54	11.42	38.84	7.12	0.00	56.93
TangentConv [189]	76.97	43.71	33.30	71.54	91.38	75.90	35.22	0.00	45.34	0.00	26.69	19.24	67.58	0.01	0.00	0.00
SPGraph [110]	85.27	44.39	37.29	69.93	94.55	88.87	32.83	12.58	15.77	15.48	30.63	22.96	56.42	0.54	0.00	44.24
SparseConv [73]	88.66	63.28	42.66	74.10	97.90	94.20	63.30	7.50	24.20	0.00	30.10	34.00	74.40	0.00	0.00	54.80
KPConv [193]	93.20	63.76	57.58	87.10	98.91	95.33	74.40	28.69	41.38	0.00	55.99	54.43	85.67	40.39	0.00	86.30

This ensured that all methods were subjected to consistent and equivalent training and evaluation procedures. To ensure a fair comparison, the training of each method was performed on the same training data split, while evaluation was performed on the same test split.

Not surprisingly, the performance of all baselines shows varying degrees of degradation, compared with that achieved in other similar aerial point cloud datasets [200]. Specifically, the recent KPConv [193] shows the best mIoU performance, but only with a mIoU score of 57.58%, which is still far from satisfactory in practice. In particular, several infrastructure-oriented semantic categories such as *rail*, *footpath*, and *bridge* are poorly segmented. Additionally, we also noticed that the category *bike* is completely misclassified by all baselines. In general, all baselines are more likely to achieve better segmentation performance in categories with simple geometrical structure and dominant proportion, such as *ground*, *vegetation*, *building* and *car*, while achieving relatively limited performance on categories such as *wall*, *bridge*, and *water*. Additionally, different baselines have vastly different performances in individual semantic categories, without a clear leader. Overall, there remain several particular challenges for selected baselines to achieve satisfactory segmentation performance in the proposed city-scale SensatUrban dataset. Motivated by this, we then dive deep into the challenges that arise from our new urban-scale dataset.

3.4 Challenges

In this section, we further analyze the key challenges to generalize existing deep segmentation models to urban-scale photogrammetry point clouds. In particular, we first identify several unique challenges from the perspective of dataset characteristics. Next, we further explore the potential solutions and perform specific

Table 3.6: Semantic segmentation results achieved by selected baselines [155, 157] with different input preparation steps. Note that, the performance of all baselines is evaluated on the original point clouds, instead of a downsampled point cloud.

	Sampling	Input sets	OA(%)	mAcc(%)	mIoU(%)	ground	veg.	building	wall	bridge	parking	rail	traffic	street	car	footpath	bike	water
PointNet [155]	Grid	Constant Number	90.57	56.30	49.69	83.55	97.67	90.66	22.56	43.54	40.35	9.29	50.74	29.58	68.24	29.27	0.00	80.55
PointNet [155]	Grid	Constant Volume	88.27	49.80	42.44	80.20	96.43	87.88	8.45	35.14	32.52	0.00	43.03	19.26	54.66	18.26	0.00	75.87
PointNet [155]	Random	Constant Number	90.34	55.17	48.49	83.47	97.51	90.89	18.55	33.31	42.82	11.85	47.95	26.83	68.37	29.12	0.00	79.71
PointNet [155]	Random	Constant Volume	88.09	48.45	41.68	79.82	96.24	87.64	5.69	27.70	34.98	0.00	42.85	13.81	54.29	20.64	0.00	78.24
PointNet++ [157]	Grid	Constant Number	93.31	66.08	58.81	86.18	98.73	95.04	63.68	51.17	55.26	0.00	60.47	46.66	82.49	40.87	0.00	84.01
PointNet++ [157]	Grid	Constant Volume	91.96	59.59	52.54	85.34	98.57	94.21	54.25	31.60	42.40	0.00	51.87	36.90	77.43	28.85	0.00	81.54
PointNet++ [157]	Random	Constant Number	93.03	64.72	58.22	85.81	98.72	94.70	59.52	57.60	52.44	0.00	58.79	46.32	82.16	38.41	0.00	82.41
PointNet++ [157]	Random	Constant Volume	91.60	56.40	50.12	84.86	98.47	93.65	51.48	15.03	39.56	0.00	50.26	36.25	75.80	28.06	0.00	78.19

experiments to verify the effectiveness. Note that, this chapter is not aiming to introduce specific new algorithms to solve all these challenges, but hopes to point out unresolved issues and provide in-depth analysis and insights, eventually stimulate the development of fine-grained urban-scale point cloud understanding.

3.4.1 Data Preparation

Considering the limited memory of modern GPUs, it is infeasible and unrealistic to directly accommodate and process urban-scale point clouds with billions of points in practice. As a result, the original point cloud data are usually partitioned into small pieces or downsampled before feed into existing neural architectures, so as to find a trade-off between the computational efficiency and segmentation accuracy.

In particular, the early works including PointNet [155], PointNet++ [157], and their variants usually first divide the large point clouds into equally-sized small blocks with partial overlap (*e.g.*, $1\text{m} \times 1\text{m}$ blocks in the S3DIS dataset [26]). However, the final segmentation performance is highly-sensitive to the input block size. Large blocks with massive points lead to an unaffordable GPU memory cost, while small blocks inevitably break the objects’ geometrical structure. Recent works such as KPConv [193] and RandLA-Net [90] resort to grid or random down-sampling at the beginning to reduce the total amount of points. Additionally, several other works [241] applied different partitioning or downsampling steps to preprocess the raw point clouds. Overall, various data preparation steps are intensively-involved in existing neural pipelines, but there are still no standard and principled preparation steps in literature, not to mention comprehensive evaluation and analysis.

To further investigate the impact of different data preparations on the final segmentation performance, we standardized a unified two-step data preprocessing framework. The detailed descriptions are as follows:

- Step 1. Reducing the redundant points in the original point clouds through downsampling at the very beginning. This can be achieved by using 1) random downsampling or 2) grid-downsampling as described in [193]. In particular, random downsampling has superior computational and memory efficiency, while grid downsampling is robust to varying point densities. Both methods can effectively reduce the total amount of points.
- Step 2. Iteratively feeding mini-batches of point subsets into the network. This can be achieved through the use of efficient space partitioning data structures, such as a KDTree, to query either constant-number point subsets or constant-volume point subsets from specific regions. Constant-number input sets were obtained by querying a fixed number of neighboring points in relation to a specific point, as described in [90]. On the other hand, constant-volume input sets were generated by cropping fixed-size point cloud chunks (*e.g.* cubes, spheres) centered on a specific point, as described in [155, 157]. The query points were randomly initialized and dynamically updated as outlined in [193].

To evaluate the impact of 4 different combinations of both Step 1 and Step 2 on the segmentation performance, we select two representative approaches PointNet [155] and PointNet++ [157] as the baselines. For a fair comparison, we set the grid size for grid downsampling as 0.2m, while the downsampling ratio is set to 1/10, so as to keep similar number of points after downsampling operation. For constant-number inputs, we implement this by using a prebuilt KDTree to query a fixed number of neighboring points of the center point as inputs. For constant volume inputs, we first crop a fixed-size volume (*e.g.*, 8m×8m block) around the center point, followed by random down(up)-sampling to align the number of different input sets. Note that, to ensure a fair comparison, the output point-wise semantic predictions are finally back-projected to the original input point clouds through the nearest neighbor assignment. The training/validation/test split follows the scheme introduced in 3.3.1.

Analysis. Table 3.6 reports the quantitative semantic segmentation scores achieved by baseline approaches with different input preparations. Table 3.7 shows the number of points left after downsampling, and the detailed time used for downsampling. It can be seen from the results that:

- The constant number input sets consistently outperform constant volume input sets in both baseline approaches. This is due to the more uniform density of

Table 3.7: Comparison of the grid downsampling and random downsampling at the beginning of data preparation.

	Number of points after sampling	Time consumption(s)
Grid downsample	220,671,929	1107
Random downsample	270,573,783	129

constant number schemes, which leads to learn a more effective feature representation.

- Both PointNet and PointNet++ exhibit slightly improved segmentation performance when using grid downsampling as opposed to random downsampling. However, the grid downsampling method consumes significantly more time (129s vs. 1107s) when evaluated on the same hardware configuration (Intel Core™ i9-10900X CPU and NVIDIA RTX 3090 GPU).

To summarize, our experiments demonstrate the importance of data preparation for the semantic segmentation performance. Although this issue has been overlooked by the community for a long time, we show that the same network architecture can bring up to 10% performance gap, when equipped with different data preparation steps. Therefore, it is desirable and encouraged to further investigate the effective data preparation schemes, especially for our urban-scale point cloud datasets.

3.4.2 Geometry vs. Appearance

Different from the point clouds acquired by the airborne LiDAR sensors [200, 167, 28, 264], the point clouds in our SensatUrban are colorized with fine-grained point-wise RGB information. Intuitively, the additional color features can provide informative appearance, further enable existing neural architectures to distinguish between heterogeneous semantic categories with similar geometrical structure (*e.g.*, grass on the ground). However, the additional color information may also introduce distractors, which in turn deteriorate the final performance.

Existing techniques usually integrated the RGB color as additional channels of the input feature map feed into the network. However, the recent ShellNet [254] learn the semantics from the pure spatial coordinates, but also achieves surprisingly good results. Overall, it remains an open question whether, and how, the color information impacts the final segmentation performance. To this end, we

Table 3.8: Evaluation of semantic segmentation performance of five selected baselines on our SensatUrban dataset with/without the usage of color information.

	OA(%)	mAcc(%)	mIoU(%)	ground	veg.	building	wall	bridge	parking	rail	traffic.	street.	car	footpath	bike	water
PointNet [155] (w/o RGB)	83.50	33.52	28.85	67.35	92.66	84.72	16.02	0.00	13.65	2.68	17.09	0.33	54.54	0.00	0.00	26.04
PointNet [155] (w/ RGB)	90.57	56.30	49.69	83.55	97.67	90.66	22.56	43.54	40.35	9.29	50.74	29.58	68.24	29.27	0.00	80.55
PointNet++ [157] (w/o RGB)	90.85	56.94	50.71	79.05	98.37	94.22	66.76	39.74	37.51	0.00	51.53	38.82	81.71	5.80	0.00	65.68
PointNet++ [157] (w RGB)	93.10	64.96	58.13	86.38	98.76	94.72	65.91	50.41	50.53	0.00	58.40	46.95	82.31	38.40	0.00	82.88
SPGraph [110] (w/o RGB)	84.81	42.12	35.29	69.60	94.18	88.15	34.55	20.53	15.83	16.34	31.44	10.54	55.01	0.98	0.00	21.57
SPGraph [110] (w RGB)	85.27	44.39	37.29	69.93	94.55	88.87	32.83	12.58	15.77	15.48	30.63	22.96	56.42	0.54	0.00	44.24
KPConv [193] (w/o RGB)	91.47	57.43	51.79	80.43	98.82	94.93	74.17	44.53	32.11	0.00	54.32	37.83	84.88	14.48	0.00	56.79
KPConv [193] (w RGB)	93.92	71.44	64.50	87.04	99.01	96.31	77.73	58.87	49.88	37.84	62.74	56.60	86.55	44.86	0.00	81.01

further conduct comparative experiments to verify the impact of the color information to the final segmentation performance. In particular, four baselines including PointNet/PointNet++ [155, 157], SPGraph [110], and KPConv [193] are selected for 8 groups of experiments. To evaluate the impact of incorporating additional appearance information on the performance of the methods, each baseline was trained using either 3D coordinates alone or in combination with RGB appearance. The experimental procedures for training and testing were maintained consistent with those described in Section 3.4.1.

Analysis. We report the quantitative results achieved by the selected five baselines with/without the usage of color in the input point clouds. We can see that:

- All of PointNet/PointNet++, and KPConv achieve significant performance improvement when the color features are utilized, compared with the use geometrical coordinates alone. It is noted that categories with significant performance improvements include *bridge*, *footpath*, and *water*, since these categories are geometrically indistinguishable.
- We also noticed that the performance improvement of SPGraph is relatively marginal (only 2%) compared with other baseline approaches. This is likely due to the homogeneous geometrical partition used in its framework, which purely relies on the geometrical structure but ignores the informative color.

Apart from the quantitative results, we also explicitly visualize the qualitative results achieved by these baselines in Figure 3.7. To summarize, our experiments highlight the importance of color information to the fine-grained understanding of urban-scale point clouds, reflecting the advantages of our SensatUrban dataset over other existing aerial point clouds datasets collected by LiDAR, such as DALES [200], NPM3D [167], and DublinCity [264]. In particular, the color information is

Table 3.9: Evaluation of semantic segmentation performance of PointNet [155] and PointNet++ [157] with different loss functions. To avoid ambiguity and repetition, we use PointNet2 instead of PointNet++ in the Table below.

	OA(%)	mAcc(%)	mIoU(%)	ground	veg.	building	wall	bridge	parking	rail	traffic.	street.	car	footpath	bike	water
PointNet+ce [155]	90.57	56.30	49.69	83.55	97.67	90.66	22.56	43.54	40.35	9.29	50.74	29.58	68.24	29.27	0.00	80.55
PointNet+wce [90]	88.13	68.05	51.24	81.01	97.12	87.87	24.46	45.76	47.78	34.93	49.82	29.58	61.28	31.78	0.00	74.67
PointNet+wce+sqrt [24]	89.72	67.97	52.35	82.87	97.33	90.42	28.32	44.94	48.39	32.07	49.58	32.63	65.11	32.59	2.60	73.71
PointNet+lovas [29]	89.58	67.50	52.53	82.74	97.27	90.28	28.11	43.89	48.53	33.58	49.68	32.21	64.01	33.05	1.46	78.13
PointNet+focal [125]	89.46	67.33	52.37	82.47	97.34	90.25	28.36	51.87	46.40	30.50	48.62	32.43	65.00	32.23	1.21	74.10
PointNet2+ce [157]	92.49	62.57	56.00	85.58	98.69	94.10	61.26	47.58	46.23	0.00	54.98	43.42	80.80	34.01	0.00	81.29
PointNet2+wce [90]	90.69	71.19	54.97	82.48	98.37	92.57	54.07	47.51	49.24	4.90	55.44	38.18	81.26	35.87	0.00	74.71
PointNet2+wce+sqrt [24]	90.80	73.80	54.66	83.16	98.39	93.03	58.44	37.77	47.27	15.80	51.26	37.00	78.71	36.26	0.34	73.19
PointNet2+lovas [29]	91.59	77.42	57.86	83.72	98.52	94.02	61.86	40.60	51.72	22.88	54.78	42.14	81.80	37.18	9.68	73.30
PointNet2+focal [125]	90.90	75.16	55.47	82.82	98.45	93.19	56.00	43.27	46.76	16.35	52.43	39.63	80.49	35.35	2.81	73.57

particularly important for distinguishing heterogeneous categories with consistent geometric structures (*e.g.*, grass on the road), enabling a higher level of semantic understanding. This also provides insights for future aerial mapping campaigns, where color information and even other spectral bands may be useful for the semantic understanding.

3.4.3 The Impact of Skewed Class Distribution

Although data preparations and the usage of color information have been considered, it is still noted that the performance of different semantic categories varies greatly. For example, all baseline methods can achieve excellent segmentation performance on *vegetation*, with IoU scores up to 99%, while completely failing in detecting rare patterns such as *bikes*. Fundamentally, this is because of the extremely imbalanced distribution of our dataset. As illustrated in Figure 3.6, the SensatUrban dataset is dominated by categories such as *ground/vegetation/building*, which are commonly appeared in urban areas of modern cities. However, categories such as *rail/bike*, despite being highly important for infrastructure-oriented applications, occur much less frequently than the prevalent categories. As consequence, the selected baselines show a biased tendency towards the prevalent categories during inference, due to the scarce occurrence of the under-represented categories.

It remains an open question to learn from training data with skewed distributions. In this thesis, we attempt to alleviate this problem from the perspective of the loss function. In particular, advanced loss functions are utilized to adaptively re-weight the contributions of each point that belongs to different categories, eventually guiding the network to achieve a more balanced performance across different categories. Specifically, by taking PointNet and PointNet++ as baselines,

Table 3.10: Cross-city generalization performance of selected baselines on our SensatUrban dataset. All baselines are trained on the *training split of Birmingham*. The top five records show the testing results on the *testing split of Birmingham*, while the bottom five rows show the scores on the *testing split of Cambridge (cross-city)*.

	OA(%)	mAcc(%)	mIoU(%)	ground	veg.	building	wall	bridge	parking	rail	traffic.	street.	car	footpath	bike	water
PointNet [155]	87.33	54.76	48.73	80.91	94.58	87.40	33.69	0.51	66.23	16.98	49.55	36.08	74.59	1.49	0.00	91.51
PointNet++ [157]	89.85	64.24	57.39	84.34	97.11	89.74	61.56	3.78	68.08	41.95	54.43	51.54	84.73	14.43	0.00	94.34
SPGraph [110]	80.13	42.87	36.95	65.75	93.33	87.24	41.28	0.00	42.69	20.94	2.28	32.05	64.06	0.00	0.00	30.76
KPConv [193]	91.44	68.41	61.65	86.00	97.66	92.90	75.07	0.91	69.74	55.50	57.94	60.73	89.48	21.44	0.00	94.13
PointNet [155]	86.06	38.56	29.70	74.94	94.57	85.38	8.62	13.42	16.47	0.00	38.64	14.27	36.96	0.09	0.00	2.75
PointNet++ [157]	89.46	44.64	36.93	77.68	97.28	91.95	54.59	0.52	15.84	0.00	42.08	29.00	67.71	0.24	0.00	3.16
SPGraph [110]	82.02	24.83	20.70	61.72	88.26	78.27	8.29	0.00	0.00	0.00	0.64	1.87	30.00	0.00	0.00	0.00
KPConv [193]	90.62	48.71	40.51	78.88	98.33	94.24	76.20	0.01	14.70	0.00	41.77	39.32	74.22	0.39	0.00	8.61

we replace the original vanilla cross-entropy loss with four off-the-shelf loss functions, including weighted cross-entropy with inverse frequency [54], or with inverse square root (sqrt) frequency [163], *Lovász*-softmax loss [29], and focal loss [125]. The experimental procedures for training and testing in this section were kept consistent with those outlined in Section 3.4.1. The only deviation from the previous setup was the use of different loss functions in this experiment.

Analysis. We report the detailed segmentation performance of two baselines achieved with five different loss functions. It can be seen that the performance of PointNet is improved when using advanced loss functions, while the mIoU score of PointNet++ is only improved when using the *Lovász*-softmax loss function. Additionally, the mAcc scores of both baselines are significantly improved, indicating that these sophisticated loss functions are effective at addressing the issue of imbalanced class distribution and resulting in more balanced segmentation performance. Notably, we also noticed that the mAcc score of PointNet++ improved by nearly 15% when using the *Lovász*-softmax loss function. In particular, the score for under-represented categories such as rail improved by more than 20%. Despite the performance in minority categories are still far from satisfactory, this improvement is considerably encouraging and suggests that further research could be conducted to fully address this research challenge.

3.4.4 Cross-City Generalization

One of the main challenges of existing deep neural architectures is how to enhance the generalization capability to unseen scenarios, especially out-of-distribution data, since neural networks are usually data hungry and tend to overfit the training data. Motivated by this, we further explore the generalization performance of

Table 3.11: Cross-city generalization performance of selected baselines on our dataset. All baselines are trained on the *training split of Cambridge*. The top five records show the testing results on the *testing split of Cambridge*, while the bottom five rows show the scores on the *testing split of Birmingham (cross-city)*.

	OA(%)	mAcc(%)	mIoU(%)	ground	veg.	building	wall	bridge	parking	rail	traffic.	street.	car	footpath	bike	water
PointNet [155]	91.16	50.02	43.61	82.83	97.89	90.93	9.54	38.34	12.07	0.00	50.60	21.42	60.74	26.18	0.00	76.42
PointNet++ [157]	93.62	62.64	55.60	85.50	98.93	95.35	63.73	59.19	24.00	0.00	59.13	40.50	79.30	38.01	0.00	79.11
SPG [110]	84.13	36.36	31.55	68.96	92.14	84.61	15.37	13.84	4.46	0.00	31.83	21.02	22.04	0.47	0.00	23.83
KPConv [193]	94.89	69.65	62.36	87.91	99.22	97.00	80.09	77.31	36.65	0.00	65.62	54.70	84.59	43.25	0.00	84.34
PointNet [155]	71.99	43.90	35.01	64.55	93.76	72.71	5.30	17.55	8.08	0.00	26.60	8.87	65.35	11.34	0.00	80.99
PointNet++ [157]	78.47	52.32	41.29	70.52	96.07	70.13	44.89	6.60	34.67	0.00	33.39	27.42	73.79	16.78	0.00	62.48
SPG [110]	71.27	28.42	22.93	57.32	84.28	76.51	12.40	8.95	0.00	0.00	20.41	10.52	14.08	0.00	0.00	13.65
KPConv [193]	86.03	61.76	50.67	78.46	97.20	81.72	55.76	40.08	64.05	0.00	44.99	38.91	80.01	17.79	0.00	59.76

existing representative baselines on our SensatUrban dataset, since our dataset is composed of data collected from different cities, hence naturally suitable for evaluating the generalization abilities. In particular, four baseline approaches are included in our generalization experiments, that is: PointNet/PointNet++ [155, 157], SPGraph [110], and KPConv [193]. We keep the same data preparation and color feature usage as mentioned in previous sections, while only change the training/test split in this section. The detailed four groups of experimental schemes are described as follows:

- Group 1: Train Birmingham/Test Birmingham. All of the four baseline approaches are only trained and tested on the training split and test split of Birmingham, respectively.
- Group 2: Generalize from Birmingham to Cambridge: All of the four baselines are trained on the training split of Birmingham, and tested on the testing split of Cambridge.
- Group 3: Train Cambridge/Test Cambridge: Analogous to group 1, all of the four baselines are only trained on the training split of Cambridge, and then tested on the testing split of the same region.
- Group 4: Generalize from Cambridge to Birmingham: the above well-trained four baseline models in group 3 are directly tested on the testing split of Birmingham.

Analysis. We report the detailed results achieved in the first two groups of experiments in Table 3.10, and the last two groups of experiments in Table 3.11. It can

be seen that the performance of all baselines shows a significant decrease (approximate 20% drop on average in mIoU scores) when generalizing the trained model to unseen urban areas in other cities, despite the data collected from different cities are actually in the same domain (*i.e.*, captured using the same sensor). This demonstrates the limited generalization capacity of selected baseline approaches. Interestingly, we also noticed that the performance of dominant semantic categories such as *vegetation* and *building* are not severely affected, while the under-presented categories including *rail* and *water* show visible performance degradation. This shows the baseline approaches are actually overfitted to the prevalent categories, while they failed to learn generalized and meaningful representation for minority categories. Overall, generalizing the trained deep segmentation model to unseen data, especially point clouds with different distributions, remains an open question in this area. Therefore, we hope our SensatUrban dataset could highlight the limited generalization capacity of existing deep neural architectures, and inspire more research to be conducted on this challenging problem.

Table 3.12: The class mapping from DALES and SensatUrban dataset to the final unified semantic categories.

Classes of DALES	Mapped class	Classes of SensatUrban
Ground	Ground	Ground, Bridge, Parking, Rail, Traffic Road, Footpath
Vegetation	Vegetation	Vegetation
Cars, Trucks	Cars	Cars
Power lines, Poles	Street furniture	Street furniture
Fences	Fences	Walls
Buildings	Buildings	Buildings
Unclassified	Unclassified (Ignored)	Bikes, Waters

3.4.5 Cross-Dataset Generalization

Another interesting question is whether the deep model trained on our SensatUrban dataset can be well generalized to other similar airborne point cloud datasets [200, 264], or vice versa. Intuitively, this task seems even more challenging than cross-city generalization, since the point clouds acquired from different cities in our dataset are inherently homogeneous. That is, the point cloud is reconstructed from sequential aerial images captured by the same camera, with the identical data process pipeline. However, point clouds in different datasets are likely to be collected by distinct acquisition sensors (*i.e.*, airborne LiDAR vs. photogrammetry

Table 3.13: Statistics of the DALES dataset and SensatUrban dataset after class mapping.

Mapped classes	DALES		SensatUrban	
	Training	Test	Training	Test
Ground	178,021,561	68,871,897	667,443,997	188,848,584
Vegetation	120,818,120	41,464,228	544,284,286	158,512,452
Cars	3,332,171	1,224,696	37,557,130	10,918,466
Street furniture	1,076,810	323,136	26,669,467	6,656,919
Fences	1,512,927	624,069	20,606,217	5,682,668
Buildings	56,908,533	23,454,294	861,977,674	164,867,233
Unclassified	6,997,560	681,571	7,262,966	4,449,979

camera) and generated by using different mapping techniques. Moreover, the data distribution, point density, geographic regions, scene contents and annotation practices may vary greatly. Albeit interesting, there are few relevant studies in the field of 3D point cloud semantic understanding.

In this thesis, we move a step forward to explore how the domain shift in different datasets affects the semantic learning of deep neural networks. Specifically, we select the recent aerial LiDAR point clouds dataset DALES [200], along with the proposed photogrammetry SensatUrban dataset, to evaluate the cross dataset generalization capacity of existing segmentation algorithms. Note that, due to the inconsistent taxonomies and annotation practice, the semantic categories (*i.e.*, 8 valid categories in DALES vs. 13 valid categories in SensatUrban) and definitions are different in these two datasets. Therefore, we first reconcile the taxonomies and map the semantic categories into the newly defined 6 consistent semantic categories, so as to properly evaluate the generalization performance across datasets. The detailed class mapping from each dataset to the unified taxonomy is shown in Table 3.12. The statistics (*i.e.*, the number of points in the training and test subset) after class mapping is reported in Table 3.13. It can be seen that the class distribution of the two datasets exhibits visible differences. Here, we select the representative PointNet as the baselines, for the evaluation of intra-dataset generalization, and cross-dataset generalization performance. Note that, the baseline approach is trained with the usage of 3D spatial coordinates only, since the color information is not available in LiDAR point clouds provided by the DALES dataset.

Analysis. Although we considered the point density and the number of input points during data preprocessing, the cross-dataset generalization performance of both two baseline approaches is still significantly lower than the intra-dataset eval-

Table 3.14: Quantitative cross-dataset generalization results were achieved by the selected baseline approach on the proposed SensatUrban dataset and the DALES dataset.

Methods	Settings	OA(%)	mIoU(%)	Ground	Vegetation	Cars	Street furniture	Fences	Buildings
PointNet [155]	DALES $\rightarrow$ DALES	94.10	59.72	94.68	86.69	16.48	73.62	0.00	86.87
	DALES $\rightarrow$ SensatUrban	74.25	30.75	89.44	55.69	0.02	0.03	0.00	39.32
	SensatUrban $\rightarrow$ SensatUrban	92.46	56.27	92.90	92.14	52.69	0.33	14.25	85.33
	SensatUrban $\rightarrow$ DALES	87.45	41.98	92.64	72.15	2.77	11.79	8.31	64.23

uation (15%+), demonstrating that domain shifts in different datasets play a key role in preventing model generalization. Future studies are encouraged to further reduce the domain gap between different point cloud datasets, especially in light of the different configurations of existing LiDAR point clouds.

3.4.6 Semantic Learning with Fewer Labels

Deep learning-based methods are hungry for massive training data [215]. For fully-supervised segmentation pipelines such as [90, 193, 155, 157], a large amount of fine-grained per-point annotations are usually required. However, it is extremely time-consuming and labor-intensive to manually annotate an urban-scale point cloud dataset with thousands of millions of points in practice. To this end, we further investigate the possibility of semantic learning with limited annotations on our SensatUrban dataset.

Table 3.15: Quantitative results achieved by PointNet [155] and PointNet++ [157] with different settings (varying number of semantic annotations). [†] means randomly a tiny fraction points of each semantic category, [‡] means randomly a tiny fraction points of the whole points.

Settings	Methods	OA(%)	mAcc(%)	mIoU(%)	ground	veg.	building	wall	bridge	parking	rail	traffic.	street.	car	footpath	bike	water
1 pt	PointNet	42.30	13.12	7.90	13.79	45.84	36.53	0.36	0.00	2.49	0.00	0.00	1.58	1.98	0.09	0.00	0.00
	PointNet++	29.52	20.82	8.74	21.48	24.52	24.80	1.81	0.00	8.99	0.00	15.56	3.52	5.91	7.02	0.02	0.03
1% annotation [†] per category	PointNet	89.58	52.87	45.95	82.92	97.40	90.07	16.12	28.83	33.82	5.86	44.95	26.98	65.50	23.15	0.00	81.81
	PointNet++	91.70	59.21	52.05	85.42	98.67	93.96	59.48	24.50	37.21	0.00	49.45	39.23	78.91	30.04	0.00	79.76
1% annotation [‡]	PointNet	89.11	50.25	43.46	82.33	97.13	89.20	11.12	15.92	34.38	0.00	42.33	23.63	63.73	26.33	0.00	78.94
	PointNet++	91.75	58.31	51.49	83.90	98.58	94.16	60.96	13.74	41.71	0.00	50.74	40.64	78.71	24.80	0.00	81.39
10% annotation [†] per category	PointNet	89.67	53.73	47.31	81.82	97.49	89.70	18.83	43.13	30.20	9.48	47.62	29.39	64.77	25.21	0.00	77.39
	PointNet++	92.60	63.24	56.67	86.08	98.73	94.65	64.68	49.87	43.77	0.00	53.70	45.51	82.29	36.20	0.00	81.27
10% annotation [‡]	PointNet	88.64	51.51	43.78	81.80	97.26	88.51	16.97	33.58	16.67	1.27	43.30	24.44	65.37	24.14	0.00	75.77
	PointNet++	92.28	60.71	53.84	85.39	98.64	94.13	60.59	25.89	46.15	0.00	53.34	42.73	80.78	30.04	0.00	82.19
Full supervision	PointNet	90.57	56.30	49.69	83.55	97.67	90.66	22.56	43.54	40.35	9.29	50.74	29.58	68.24	29.27	0.00	80.55
	PointNet++	93.10	64.96	58.13	86.38	98.76	94.72	65.91	50.41	50.53	0.00	58.40	46.95	82.31	38.40	0.00	82.88

Inspired by the weak supervision setting proposed in [234], we have conducted six groups of experiments by training all baselines with different forms of semantic annotations (*i.e.*, weak supervisions) in our dataset. We keep the same data preparation and color feature usage as mentioned in previous sections, while only

Table 3.16: Quantitative results achieved by using OcCo [207], Jigsaw [169] and Random (Rand) initialization on the SensatUrban dataset, based on PointNet [155], PCN [245] and DGCNN [213] encoders. Note that, all the initialized weights are obtained by pre-training on the ModelNet40 [224], since these techniques are mainly designed for object-level classification and segmentation.

	OA(%)	mAcc(%)	mIoU(%)	ground	veg.	building	wall	bridge	parking	rail	traffic.	street.	car	footpath	bike	water
PointNet [155]	86.29	53.33	45.10	80.05	93.98	87.05	23.05	19.52	41.80	3.38	43.47	24.20	63.43	26.86	0.00	79.53
PointNet-Jigsaw [169]	87.38	56.97	47.90	83.36	94.72	88.48	22.87	30.19	47.43	15.62	44.49	22.91	64.14	30.33	0.00	77.88
PointNet-OcCo [207]	87.87	56.14	48.50	83.76	94.81	89.24	23.29	33.38	48.04	15.84	45.38	24.99	65.00	27.13	0.00	79.58
PCN [245]	86.79	57.66	47.91	82.61	94.82	89.04	26.66	21.96	34.96	28.39	43.32	27.13	62.97	30.87	0.00	80.06
PCN-Jigsaw [169]	87.32	57.01	48.44	83.20	94.79	89.25	25.89	19.69	40.90	28.52	43.46	24.78	63.08	31.74	0.00	84.42
PCN-OcCo [207]	86.90	58.15	48.54	81.64	94.37	88.21	25.43	31.54	39.39	22.02	45.47	27.60	65.33	32.07	0.00	77.99
DGCNN [213]	87.54	60.27	51.96	83.12	95.43	89.58	31.84	35.49	45.11	38.57	45.66	32.97	64.88	30.48	0.00	82.34
DGCNN-Jigsaw [169]	88.65	60.80	53.01	83.95	95.92	89.85	30.05	43.59	46.40	35.28	49.60	31.46	69.41	34.38	0.00	80.55
DGCNN-OcCo [207]	88.67	61.35	53.31	83.64	95.75	89.96	29.22	41.47	46.89	40.64	49.72	33.57	70.11	32.35	0.00	79.74

change the semantic annotations of training splits in this section. For simplicity, we only adopt PointNet [155] and PointNet++ [157] as baseline networks in the following groups of experiment:

- Only 1 point annotated per category in each point cloud.
- Only 1% points annotated per category in each point cloud.
- Only 1% points annotated in each point cloud (randomly).
- 10% points annotated per category in each point cloud.
- 10% points annotated in each point cloud (randomly).
- 100% (all) points annotated in each point cloud.

Analysis. Table 3.15 shows the detailed quantitative results achieved by three baselines under different settings. It can be seen that:

- Surprisingly, both PointNet and PointNet++ can achieve comparable performance with their fully-supervised counterpart, even when training with a small fraction of labeled points (*e.g.*, 1% or 10%). This implies that the existing per-point annotations may exist with large information redundancy, and it is possible to learn semantics with limited annotations.
- The performance of all baselines in group 2 and group 4 are better than group 3 and group 5, demonstrating that randomly annotating a tiny fraction (*e.g.*, 1%, 10%) of all points is inferior to randomly annotating a tiny fraction of points in each semantic category. Although randomly annotating a tiny fraction of all points is more practical and feasible.

- The segmentation performance of all baselines in group 1 (*i.e.*, with 1 point annotation) are far from satisfactory. Basically, the networks cannot converge, primarily because the supervision information is extremely insufficient. However, this is also one of the simplest and cheapest ways of annotation in practice. More studies should be conducted in this direction to further improve the performance.

Thanks to the availability of several large-scale point cloud datasets, the community can always assume that the amount of labeled training data is sufficient. However, we have demonstrated that comparable performance can also be achieved using the same architecture with limited semantic annotations, highlighting the great potential of weakly-supervised semantic segmentation frameworks. This motivates us to further investigate how to achieve better performance under limited annotation, and how to choose the best annotation strategy under fixed budgets.

3.4.7 Self-supervised Pre-Training on 3D Point Clouds

Pre-training a network on a rich source set in a self-supervised or unsupervised way has been demonstrated to be highly effective for high-level downstream tasks (*e.g.*, segmentation, detection) in several 2D vision tasks [43, 81]. However, self-supervised pre-training on 3D point clouds is still in its infancy, only a handful of recent works [169, 207, 227, 252, 84, 154] have started to explore self-supervised learning on unstructured 3D point clouds. In particular, all existing methods are still pre-trained on the object-level datasets (*e.g.*, ModelNet40 [224]) or indoor scene-level datasets (*e.g.*, ScanNet [55]). Considering the urban-scale property of SensatUrban, it is particularly suitable for verifying the effectiveness of the existing pretraining strategy on our dataset.

To this end, we conducted three groups of experiments on our SensatUrban dataset to compare the performance of:

- Pre-training with occlusion completion [207].
- Pre-training with context prediction (jigsaw) [169].
- Training from scratch.

For simplicity, we faithfully follow the three baseline networks used in their original paper [207, 169], including PointNet [155], PCN [245], and DGCNN [213]. The detailed experimental results are shown in Table 3.16.

Analysis. From the results in Table 3.16 we can see that, although both baseline networks are purely pre-trained on the object-level point clouds in ModelNet40, the fine-tuning models can still achieve a certain performance improvement on our dataset. In particular, the performance of several minority categories, such as *rail* and *bridge*, has a significant performance improvement (up to nearly 10%), primary because the pre-trained models are less prone to overfitting to the majority categories, compared to directly training from scratch.

This further demonstrates the feasibility and potential of the self-supervised pre-training paradigm. However, the existing pre-training frameworks [207, 169] are still limited to object-level point clouds, and it is non-trivial to be extended to large-scale point clouds. On the other hand, most of the existing pre-training schemes are based on auxiliary (pre-text) tasks. It is worth investigating how to leverage contrastive learning to achieve better performance on 3D point clouds. Finally, to further facilitate the research in this research area, we also release the unlabeled York point clouds, encouraging further research exploration on this part of the data.

3.5 Discussion

The previous section outlined several unique challenges that pose barriers to the semantic understanding of large-scale 3D point clouds. In this discussion section, we aim to delve deeper into these challenges and provide a systematic examination of the findings from our experiments and analyses. By analyzing the results in detail, we hope to gain a better understanding of the challenges and their potential solutions, and contribute to the advancement of knowledge in the field of large-scale 3D point cloud semantic understanding. Specifically,

- Data preparation is a crucial aspect in achieving semantic understanding of large-scale 3D point clouds. Our analysis revealed that different data preparation methods can have a significant impact on the performance of the results. Despite its importance, this aspect has often been neglected and sometimes mis-attributed to improvements in network design. Our findings highlight the need for further research in this area, with the goal of developing

a standard and unified scheme for data preparation. This would allow for more meaningful comparisons and fair evaluations of different methods.

- Both geometric information and appearance information plays a vital role in the field of 3D semantic understanding. Our analysis showed that compared to airborne LiDAR-based point clouds, photogrammetry-based point clouds contain abundant geometric information, enabling the distinction of heterogeneous categories with similar geometric structures, and providing a more comprehensive understanding of urban semantics. However, further exploration is still worth exploring in utilizing these two types of information more effectively. This raises an important direction for future research, exploring methods to more optimally combine and leverage the strengths of both geometric and appearance information.
- The skewed class distribution has been a significant challenge in the field of semantic segmentation. Our analysis suggests that utilizing suitable loss functions can result in improved performance for underrepresented categories. However, a complete solution to this issue remains elusive. In future work, it would be valuable to investigate potential solutions to further address the problem of imbalanced class distribution, such as resampling, ensemble, and adaptive weighting. These approaches have shown promise in other domains and may prove effective in addressing the skewed class distribution in semantic segmentation of large-scale 3D point clouds.
- In terms of generalization, our findings indicate that existing algorithms for point cloud semantic segmentation struggle with both cross-city and cross-dataset generalization. This highlights the importance of addressing the issue of generalization in future work in the field. A promising direction would be to investigate methods to improve the generalizability of models, so that they can effectively handle changes in data distribution and domain discrepancy. This could involve the development of more robust algorithms, the utilization of transfer learning techniques, or other innovative solutions. By addressing this challenge, we can improve the reliability and practicality of point cloud semantic segmentation algorithms for real-world applications.
- Regarding learning with few labels, our experimental results demonstrate the great potential of utilizing weakly supervised and self-supervised learning frameworks in the field of 3D point clouds. This direction holds signifi-

cant practical significance, as it has the potential to greatly reduce the labeling cost of massive 3D points. The field of weakly supervised learning has shown impressive achievements in 2D images and these advancements can be leveraged and adapted to the realm of 3D point clouds, potentially resulting in improved performance of algorithms in the presence of weak (self) supervision.

The proposed SensatUrban dataset represents a considerable advancement in the field of photogrammetric point cloud datasets. However, it is important to acknowledge its limitations in order to guide future research efforts. Despite being the largest publicly available photogrammetric point cloud dataset, the dataset is limited in that it does not include instance annotations. This is a challenging task as obtaining point-wise instance labels is extremely labor-intensive. Nonetheless, this is an important area for future investigation [41]. Additionally, the dataset is constructed from sequential aerial images captured by a single sensor (*i.e.*, camera). It would be interesting to further investigate the same-source data acquired by different sensors. For example, the data acquired from both a camera and a LiDAR system integrated on the same UAV platform [101].

3.6 Summary and Outlook

This chapter introduces SensatUrban: an urban-scale photogrammetric point cloud dataset composed of 7.6 km^2 of urban areas in three UK cities, and nearly 3 billion richly annotated points (each with one of the 13 semantic categories). A comprehensive benchmark is also built based on this dataset and a number of selected representative baselines. In particular, extensive comparative experiments have revealed several challenges in generalizing existing semantic segmentation methods to urban-scale point clouds, including how to conduct data preparation, whether and how to utilize the color information, how to tackle the extremely imbalanced class distribution, generalizing to unseen scenarios, and the potential of weakly/self-supervised learning techniques. Besides, extensive benchmarking results are conducted and in-depth analysis is also provided. In the future, we will further increase the scale and richness (*i.e.*, instance annotation, corresponding 2D images) of our SensatUrban dataset. We hope that our SensatUrban dataset could be an immensely useful resource and a canonical benchmark to related research

communities including 3D computer vision, earth vision, and remote sensing, inspiring and supporting future advancing research in related areas.

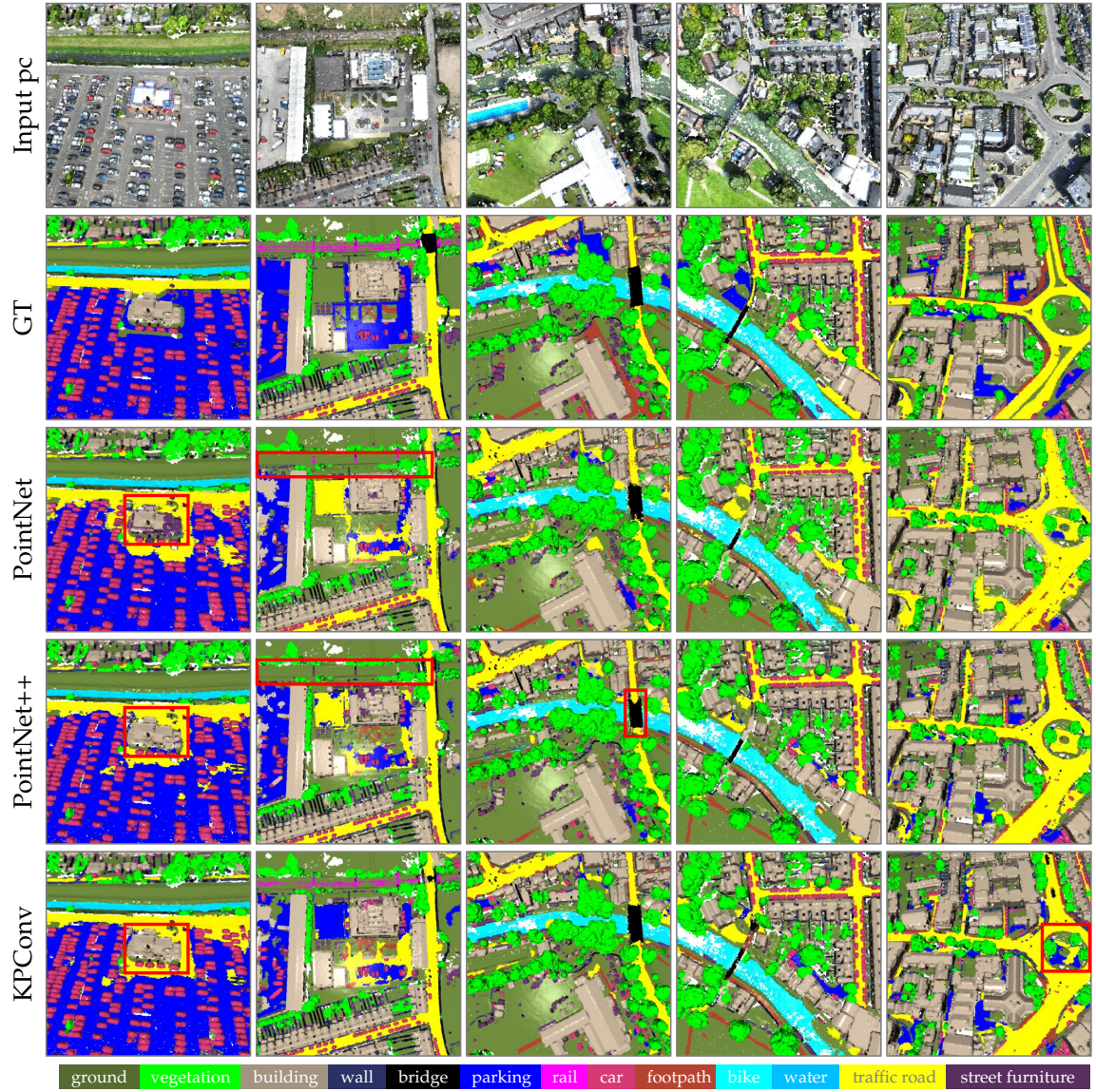


Figure 3.7: Qualitative results of PointNet [155], PointNet++ [157], and KPConv [193] on the test set of SensatUrban dataset. The black dashed box highlights the inconsistency predictions with the ground-truth label.

Chapter 4

Learning to Segment Large-Scale 3D Point Clouds

4.1 Introduction

In the previous chapter, we discussed the pressing need for semantic understanding of large-scale 3D point clouds, particularly in the context of urban-scale 3D datasets. We highlighted the unique challenges associated with this task and the importance of developing efficient and effective learning algorithms to tackle these challenges. In this chapter, we delve deeper into the topic and explore strategies for learning efficient 3D representations and achieving effective semantic segmentation of large-scale 3D point clouds. These capabilities hold significant value for real-time intelligent systems, including applications in autonomous driving [69] and augmented reality [85].

Through a comprehensive review of the relevant literature, the existing point-based semantic segmentation algorithms can be generally categorized as: 1) neighbouring feature pooling [157, 120, 94, 255, 254], 2) graph message passing [213, 171, 203, 208, 39, 98, 128, 174], 3) kernel-based convolution [179, 93, 222, 114, 102, 108, 193, 140, 212, 196, 229], and 4) attention-based aggregation [228, 247, 240, 151]. Despite their impressive results in object recognition and semantic segmentation, these approaches are limited in their ability to handle large-scale 3D point clouds (*e.g.*, millions of points and up to hundreds of meters). The limitations stem from three main aspects: 1) The commonly used point-sampling methods of these networks are either computationally expensive or memory inefficient. For example, the widely employed farthest-point sampling [157] takes over 200 seconds to sample 10% of 1 million points. 2) Most existing local feature learners usually rely on

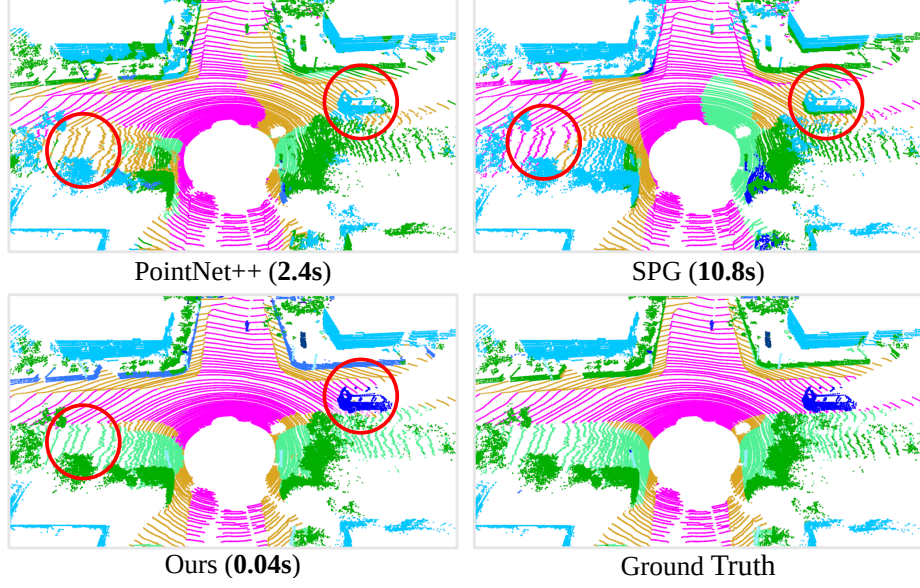


Figure 4.1: Semantic segmentation results of PointNet++ [157], SPG [110] and our approach on SemanticKITTI [28]. Our RandLA-Net takes only 0.04s to directly process a large point cloud with 10^5 points over $150 \times 130 \times 10$ m in 3D space, which is up to $200\times$ faster than SPG. Red circles highlight the superior segmentation accuracy of our approach.

computationally expensive kernelisation or graph construction, thereby being unable to process massive number of points. 3) For a large-scale point cloud, which usually consists of hundreds of objects, the existing local feature learners are either incapable of capturing complex structures, or do so inefficiently, due to their limited size of receptive fields.

A handful of recent works have started to tackle the task of directly processing large-scale point clouds. SPG [110] preprocesses the large point clouds as super graphs before applying neural networks to learn per super-point semantics. However, the computational cost of these preprocessing steps renders SPG infeasible for real-time applications. Both FCPN [160] and PCT [42] combine voxelization and point-level networks to process massive point clouds. However, they still resort to partitioning the point clouds into smaller blocks, compromising the overall performance.

In this chapter, we aim to design a memory and computationally efficient neural architecture, which is able to directly process large-scale 3D point clouds in a single pass, without requiring pre/post-processing steps such as voxelization, block partitioning or graph construction. However, this task is extremely challenging as it requires: 1) a memory and computationally efficient sampling approach

to progressively downsample large-scale point clouds to fit in the limits of current GPUs, and 2) an effective local feature learner to progressively increase the receptive field size to preserve complex geometric structures.

To achieve these goals, we first systematically analyse the relative merits of different sampling approaches in Section 4.3.1, where reveals that conventional sampling approaches can hinder scalability and impede real-time processing for large-scale point clouds. As a result, we identify random sampling as an appropriate strategy for large-scale point cloud processing due to its efficiency and scalability. However, the use of random sampling may result in a loss of critical information, particularly for objects with sparse points. To counter the potentially detrimental impact of random sampling, we design a new local feature aggregation module in Section 4.3.2, which is capable of effectively learning complex local structures by progressively increasing the receptive field size in each neural layer. In particular, for each 3D point, we firstly introduce a local spatial encoding (LocSE) unit to explicitly preserve local geometric structures. Secondly, we leverage attentive pooling to automatically keep the useful local features. By stacking multiple LocSE and attentive pooling units as a dilated residual block, we significantly enhance the effective receptive field for each point. Note that all these neural components are implemented as shared MLPs, and are therefore remarkably memory and computational efficient.

Overall, being built on the principles of simple **random** sampling and an effective local feature **aggregator**, our efficient neural architecture, named **RandLA-Net**, not only is up to $200\times$ faster than existing approaches on large-scale point clouds, but also surpasses the state-of-the-art semantic segmentation methods on Semantic3D [76], SemanticKITTI [28] and Toronto-3D [185] benchmarks. Figure 4.1 shows qualitative results of our approach. Our key contributions are:

- A comprehensive analysis of existing sampling approaches, highlighting the suitability of random sampling for efficient learning on large-scale point clouds.
- The development of a novel local feature aggregation module, which effectively preserves complex local structures by progressively increasing the receptive field for each point.
- The demonstration of significant memory and computational savings for semantic segmentation of large-scale 3D point clouds, as well as superior performance over state-of-the-art methods on multiple large-scale benchmarks.

The content of this chapter was first published in [90] and the official implementation of the proposed RandLA-Net is available as open-source code library, which can be accessed at this link¹. This work has been recognized as one of the “most influential papers in CVPR 2020” according to PaperDigest².

4.2 Overview

As illustrated in Figure 4.2, given a large-scale point cloud with millions of points which could span hundreds of meters, to process it with a deep neural network inevitably requires those points to be progressively and efficiently downsampled in each neural layer, without losing the useful point features. In our RandLA-Net, we propose to use the simple and fast approach of random sampling to greatly decrease point density, whilst applying a carefully designed local feature aggregator to retain prominent features. This allows the entire network to achieve an excellent trade-off between efficiency and effectiveness.

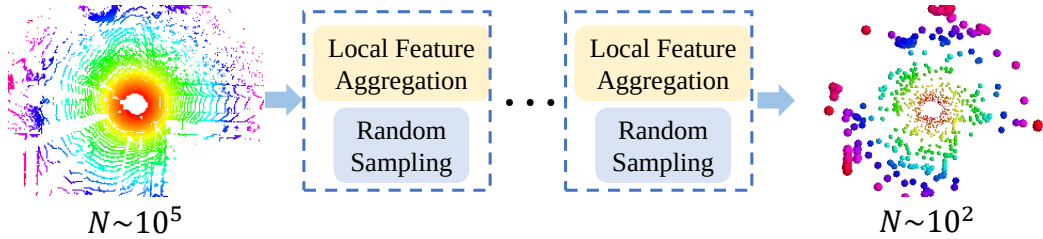


Figure 4.2: In each layer of RandLA-Net, the large-scale point cloud is significantly downsampled, yet is capable of retaining features necessary for accurate segmentation.

4.3 Method

4.3.1 The quest for efficient sampling

To determine the most suitable method for processing large-scale point clouds, we provide an empirical analysis and comparison of different sampling approaches. In particular, the relative merits and complexity of the commonly used sampling strategies are shown in Table 4.1.

¹<https://github.com/QingyongHu/RandLA-Net>

²<https://www.paperdigest.org/2022/02/most-influential-cvpr-papers-2022-02/>

Table 4.1: Comparison of representative sampling methods for processing point clouds. FPS: Farthest Point Sampling; IDIS: Inverse Density Importance Sampling; PDS: Poisson Disk Sampling; RS: Random Sampling; GS: Generator-based Sampling; CRS: Continuous Relaxation based Sampling; PGS: Policy Gradient-based Sampling. Note that, ¹the complexity denotes the actual computational complexity when sampling M points from a large-scale point cloud $\mathbf{P}$ with N points, K denotes the number of nearest neighbours. ²The time consumption is computed for sampling 10% points from 1 million points. We use the same hardware as in Section 3.5, unless specified otherwise.

Type	Method	Complexity ¹	Time (seconds) ²	Description
Heuristic Sampling	FPS [157]	$\mathcal{O}(M^2N)$	200	Farthest Point Sampling iteratively returns a reordering of the metric space $\{p_1 \dots p_m \dots p_M\}$, such that each p_m is the farthest point from the first $m - 1$ points. It has good coverage of points but high computational complexity.
	IDIS [74]	$\mathcal{O}((K + N)\log N)$	10	Inverse Density Importance Sampling reorders all N points according to the density of each point, after which the top M points are selected. The density is approximated by calculating the summation of the distances between the point and its nearest K points. It can control density, but sensitive to outliers and noise.
	PDS [34]	$\mathcal{O}(MN)$	8	Poisson Disk Sampling samples points from a set of N points with blue noise characteristics <i>i.e.</i> , points are sampled from a Poisson disk distribution, where all samples are at least a certain distance r apart.
	RS	$\mathcal{O}(M)$	0.004	Random Sampling uniformly selects M points from the original N points, each point has the same probability to be selected. It is agnostic to the total number of input points, <i>i.e.</i> , it is constant-time and hence is inherently scalable.
Learning-based Sampling	GS [58]	-	1200	Generator based Sampling learns to generate a small set of points to approximately represent the original point set. FPS matching is required during inference.
	CRS [22]	Estimated 300 GB GPU memory footprint		Continuous Relaxation based Sampling uses the reparameterization trick to relax the sampling operation to a continuous domain for end-to-end training. A large weight matrix is required when sampling all the new points simultaneously with one pass matrix multiplication, leading to an unaffordable memory cost.
	PGS [232]	$C_{10^6}^{10^5}$ exploration space		Policy Gradient-based Sampling formulates the sampling operation as a Markov decision process. It sequentially learns a probability distribution to sample the points. However, the learned probability has high variance due to the extremely large exploration space when the point cloud is large.

The implementation details of different sampling approaches are summarized as follows. To sample M points (point features) from a large-scale point cloud P with N points (point features):

1. *Farthest Point Sampling (FPS)*: We follow the implementation³ provided by PointNet++ [157], which is also widely used in [122, 222, 131, 40, 255]. In particular, FPS is implemented as an operator running on GPU.
2. *Inverse Density Importance Sampling (IDIS)*: Given a point p_i , its density ρ is approximated by calculating the summation of the distances between p_i and its nearest t points [74]. Formally:

$$\rho(p_i) = \sum_{j=1}^t \left\| p_i - p_i^j \right\|, p_i^j \in \mathcal{N}(p_i) \quad (4.1)$$

where p_i^j represents the coordinates (i.e. x-y-z) of the j^{th} point of the neighbour points set $\mathcal{N}(p_i)$, t is set to 16. All the points are ranked according to the inverse density $\frac{1}{\rho}$ of points. Finally, the top M points are selected.

3. *Poisson Disk Sampling (PDS)*: We implement Fast Poisson Disk Sampling [34] using the PDAL library⁴ for sampling M points with Poisson disk property from an input cloud with N points, given a Poisson radius r . We implement simple batch parallelisation.
4. *Random Sampling (RS)*: This is implemented with the python numpy package. Specifically, we first use the numpy function `numpy.random.choice()` to generate M indices. We then gather the corresponding spatial coordinates and per-point features from point clouds by using these indices.
5. *Generator-based Sampling (GS)*: The implementation follows the code⁵ provided by [58]. We first train a ProgressiveNet [58] to transform the raw point clouds into ordered point sets according to their relevance to the task. After that, the first M points are kept, while the rest is discarded.
6. *Continuous Relaxation based Sampling (CRS)*: CRS is implemented with the self-attended gumbel-softmax sampling [22][240]. Given a point feature set $P \in \mathbb{R}^{N \times (d+3)}$ with 3D coordinates and per point features, we firstly estimate

³<https://github.com/charlesq34/pointnet2>

⁴<https://pdal.io/>

⁵https://github.com/orendv/learning_to_sample

a probability score vector $\mathbf{s} \in \mathbb{R}^N$ through a score function parameterized by a MLP layer, i.e., $\mathbf{s} = \text{softmax}(\text{MLP}(\mathbf{P}))$, which learns a categorical distribution. Then, with the Gumbel noise $\mathbf{g} \in \mathbb{R}^N$ drawn from the distribution $\text{Gumbel}(0, 1)$. Each sampled point feature vector $\mathbf{y} \in \mathbb{R}^{d+3}$ is calculated as follows:

$$\mathbf{y} = \sum_{i=1}^N \frac{\exp((\log(\mathbf{s}^{(i)}) + \mathbf{g}^{(i)})/\tau) \mathbf{P}^{(i)}}{\sum_{j=1}^N \exp((\log(\mathbf{s}^{(j)}) + \mathbf{g}^{(j)})/\tau)}, \quad (4.2)$$

where $\mathbf{s}^{(i)}$ and $\mathbf{g}^{(i)}$ indicate the i^{th} element in the vector $\mathbf{s}$ and $\mathbf{g}$ respectively, $\mathbf{P}^{(i)}$ represents the i^{th} row vector in the input matrix $\mathbf{P}$. $\tau > 0$ is the annealing temperature. When $\tau \rightarrow 0$, Equation 4.2 approaches the discrete distribution and samples each row vector in $\mathbf{P}$ with the probability $p(\mathbf{y} = \mathbf{P}^{(i)}) = \mathbf{s}^{(i)}$.

7. *Policy Gradients based Sampling (PGS)*: Given a point feature set $\mathbf{P} \in \mathbb{R}^{N \times (d+3)}$ with 3D coordinates and per point features, we first predict a score $\mathbf{s}$ for each point, which is learnt by an MLP function, i.e., $\mathbf{s} = \text{softmax}(\text{MLP}(\mathbf{P})) + \mathbf{f}$, where $\mathbf{f} \in \mathbb{R}^N$ is a zero-mean Gaussian noise with the variance Σ for random exploration. After that, we sample M vectors in $\mathbf{P}$ with the top M scores. Sampling each point/vector can be regarded as an independent action and a sequence of them form a sequential Markov Decision Process (MDP) with the following policy function π :

$$a_i \sim \pi(a | \mathbf{P}^{(i)}; \theta, \mathbf{s}) \quad (4.3)$$

where a_i is the binary decision of whether to sample the i^{th} vector in $\mathbf{P}$ and θ is the network parameter of the MLP. Hence to properly improve the sampling policy with an underivable sampling process, we apply REINFORCE algorithm [183] as the gradient estimator. The segmentation accuracy R is applied as the reward value for the entire sampling process as $\mathcal{J} = R$. It is optimized with the following estimated gradients:

$$\frac{\partial \mathcal{J}}{\partial \theta} \approx \frac{1}{M} \sum_{m=1}^M \sum_{i=1}^N \frac{\partial}{\partial \theta} \log \pi(a_i | \mathbf{P}^{(i)}; \theta, \Sigma) \times (R - b^c - b(\mathbf{P}^{(i)})), \quad (4.4)$$

where M is the batch size, b^c and $b(\mathbf{P}^{(i)})$ are two control variates [145] for alleviating the high variance problem of policy gradients.

From the summary in Table 4.1, we can see that FPS, IDIS and GS are too computationally expensive to be applied for large-scale point clouds. CRS approaches

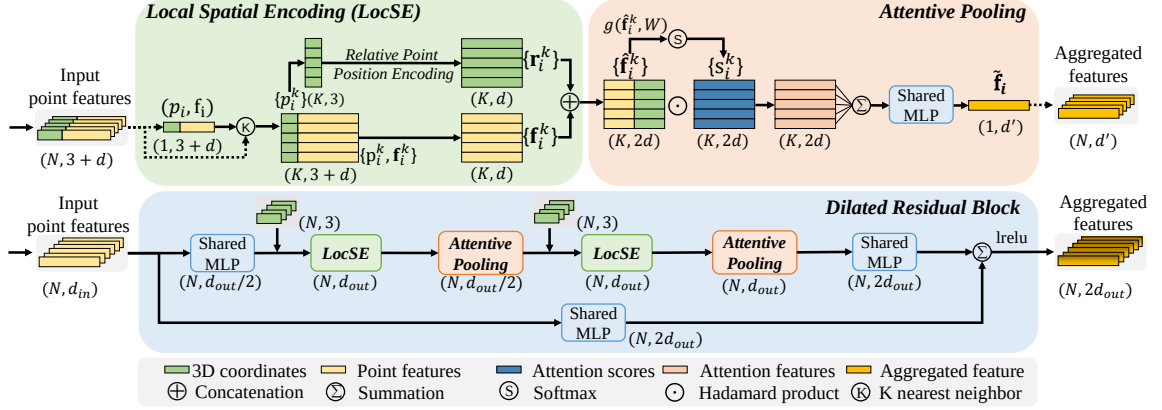


Figure 4.3: The proposed local feature aggregation module. The top panel shows the local spatial encoding block that extracts features, and the attentive pooling mechanism that weights the important neighbouring features, based on the local context and geometry. The bottom panel shows how two of these components are chained together, to increase the receptive field size, within a residual block.

have an excessive memory footprint and PGS is hard to learn. PDS is relatively faster, but has worse performance, as shown in the ablation study. By contrast, random sampling has the following two advantages: 1) it is remarkably computational efficient as it is agnostic to the total number of input points, 2) it does not require extra memory for computation. Therefore, we believe that random sampling is more suitable than other approaches to efficiently process large-scale point clouds. However, random sampling may result in many useful point features being dropped. To overcome it, we propose a powerful local feature aggregation module as presented in the next section.

4.3.2 Local Feature Aggregation

As shown in Figure 4.3, our local feature aggregation module is applied to each 3D point in parallel and it consists of three neural units: 1) local spatial encoding (LocSE), 2) attentive pooling, and 3) dilated residual block.

4.3.2.1 Local Spatial Encoding

Given a point cloud P together with per-point features (e.g., raw RGB, or intermediate learned features), this local spatial encoding unit explicitly encodes the spatial positional relationships of each point in the local neighborhoods, instead of the pure x-y-z coordinates of all neighbouring points [157, 222], such that the corresponding point features are always aware of their relative spatial locations. This

allows the LocSE unit to explicitly observe the local geometric patterns, thus eventually benefiting the entire network to effectively learn complex local structures. In particular, this unit includes the following steps. Note that, all the following steps are executed parallel for all input points.

Finding Neighbouring Points. For the i^{th} point, its neighbouring points are firstly gathered by the simple K -nearest neighbours (KNN) algorithm for efficiency. The KNN is based on point-wise Euclidean distances.

Relative Point Position Encoding. For each of the nearest K points $\{p_i^1 \cdots p_i^k \cdots p_i^K\}$ of the center point p_i , we explicitly encode the relative point position as follows:

$$\mathbf{r}_i^k = MLP\left(p_i \oplus p_i^k \oplus (p_i - p_i^k) \oplus ||p_i - p_i^k||\right) \quad (4.5)$$

where p_i and p_i^k are the absolute x-y-z positions of points, $\oplus$ is the concatenation operation, and $||\cdot||$ calculates the Euclidean distance between the neighbouring and center points. The concated point positions are further encoded by a shared MLP layer, where the trainable parameters are shared across different local neighborhoods. Interestingly, we are aware that $\mathbf{r}_i^k$ seems encoded from redundant point positions, but we found that this tends to aid the network to learn local features and obtains good performance in practice, with almost negligible computational overheads. Please refer to the experimental section for more details.

Point Feature Augmentation. For each neighbouring point p_i^k , the encoded relative point positions $\mathbf{r}_i^k$ are concatenated with its corresponding point features $\mathbf{f}_i^k$, obtaining an augmented feature vector $\hat{\mathbf{f}}_i^k$. For simplicity, we keep the feature vectors $\mathbf{r}_i^k$ and $\mathbf{f}_i^k$ with the same dimension in the implementation, but they are flexible and can have different dimensions.

$$\hat{\mathbf{f}}_i^k = p_i^k \oplus \mathbf{r}_i^k \quad (4.6)$$

Eventually, the output of the LocSE unit is a new set of neighbouring features $\hat{\mathbf{F}}_i = \{\hat{\mathbf{f}}_i^1 \cdots \hat{\mathbf{f}}_i^k \cdots \hat{\mathbf{f}}_i^K\}$, which explicitly encodes the local geometric structures for the center point p_i . We notice that the recent work [131] also uses point positions to improve the performance of semantic segmentation. However, the positions are used to learn point scores in [131], while our LocSE explicitly encodes the relative positions to augment the neighbouring point features. Note that all operations of our local feature aggregation module are shared in each local neighborhood, hence the total number of parameters being independent of the number of input points.

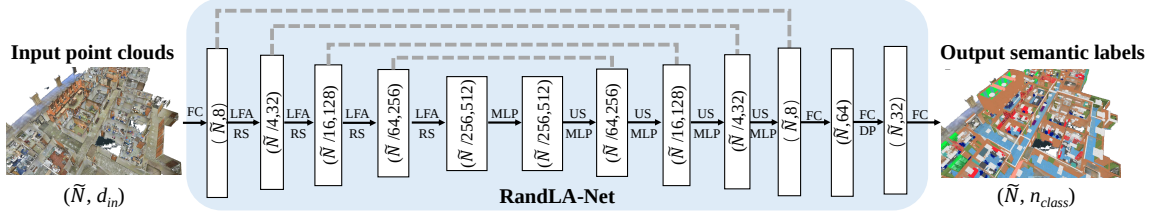


Figure 4.4: The detailed architecture of our RandLA-Net. $(\tilde{N}, D)$ represents the number of points and feature dimension respectively. FC: Fully Connected layer, LFA: Local Feature Aggregation, RS: Random Sampling, MLP: shared Multi-Layer Perceptron, US: Up-sampling, DP: Dropout.

4.3.2.2 Attentive Pooling

This neural unit is used to aggregate the set of neighbouring point features $\hat{\mathbf{F}}_i$. Existing works [157, 122] typically use max/mean pooling to hard integrate the neighbouring features, resulting in the majority of the information being lost. By contrast, we turn to the powerful attention mechanism to automatically learn important local features. In particular, inspired by [239], our attentive pooling unit consists of the following steps.

Computing Attention Scores. Given the set of local features $\hat{\mathbf{F}}_i = \{\hat{\mathbf{f}}_i^1 \dots \hat{\mathbf{f}}_i^K \dots \hat{\mathbf{f}}_i^K\}$, we design a shared function $g(\cdot)$ to learn a unique attention score for each feature. Essentially, the function $g(\cdot)$ consists of a shared MLP followed by *softmax*. It is formally defined as follows:

$$\mathbf{s}_i^k = g(\hat{\mathbf{f}}_i^k, \mathbf{W}) \quad (4.7)$$

where $\mathbf{W}$ is the learnable weights of a shared MLP.

Weighted Summation. The learned attention scores can be regarded as a soft mask which automatically selects the important features. Formally, these features are weighted summed as follows:

$$\tilde{\mathbf{f}}_i = \sum_{k=1}^K (\hat{\mathbf{f}}_i^k \odot \mathbf{s}_i^k) \quad (4.8)$$

where $\odot$ is the element-wise product. To summarize, given the input point cloud $\mathbf{P}$, for the i^{th} point p_i , our LocSE and Attentive Pooling units learn to aggregate the geometric patterns and features of its K nearest points, and finally generate an informative feature vector $\tilde{\mathbf{f}}_i$.

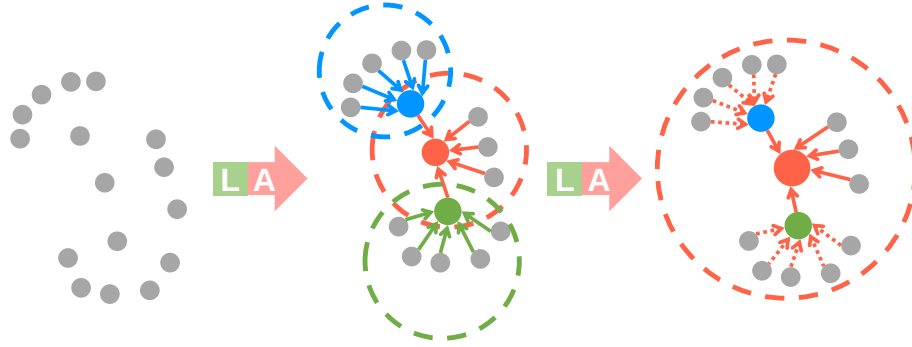


Figure 4.5: Illustration of the dilated residual block which significantly increases the receptive field (dotted circle) of each point, colored points represent the aggregated features. L: Local spatial encoding, A: Attentive pooling.

4.3.2.3 Dilated Residual Block

Since the large point clouds are going to be substantially downsampled, it is desirable to significantly increase the receptive field for each point, such that the geometric details of input point clouds are more likely to be preserved, even if some points are dropped. As shown in Figure 4.3, inspired by the successful ResNet [82] and the effective dilated networks [60], we stack multiple LocSE and Attentive Pooling units to achieve the dilation of point receptive fields, and add a skip connection to achieve residual learning.

To further illustrate the capability of our dilated residual block, Figure 4.5 shows that the red 3D point observes K neighbouring points after the first LocSE/Attentive Pooling operation, and then is able to receive information from up to K^2 neighbouring points i.e. its two-hop neighbourhood after the second. This is a cheap way of dilating the receptive field and expanding the effective neighbourhood through feature propagation. Theoretically, the more units we stack, the more powerful this block as its sphere of reach becomes greater and greater. However, more units would inevitably sacrifice the overall computation efficiency. In addition, the entire network is likely to be over-fitting. In our RandLA-Net, we simply stack two sets of LocSE and Attentive Pooling as the standard residual block, achieving a satisfactory balance between efficiency and effectiveness.

Overall, our local feature aggregation module is designed to effectively preserve complex local structures via explicitly considering neighbouring geometries and significantly increasing receptive fields. Moreover, this module only consists of feed-forward MLPs, thus being computationally efficient. In addition, it is worth mentioning that the proposed local feature aggregation module is also

permutation-equivariant to unordered 3D point clouds. This is because this module is performed in the local neighborhood of each point, hence agnostic to the order of the points. In other words, no matter how the order of the points changes, the local neighborhood of each point in the 3D space will not change, hence the learned feature through our local feature aggregation module keep the same. This property also respects the permutation invariance of 3D point clouds.

4.3.3 Network Architecture

Figure 4.4 shows the detailed architecture of RandLA-Net, which stacks multiple local feature aggregation modules and random sampling layers. The network follows the widely-used encoder-decoder architecture with skip connections. The input point cloud is first fed to a shared MLP layer to extract per-point features. Four encoding and decoding layers are then used to learn features for each point. At last, three fully-connected layers and a dropout layer are used to predict the semantic label of each point. By using dilated residual blocks for local feature aggregation and combining them with hierarchical downsampling, each point can learn a local geometric feature representation from point-level attributes, similar to traditional hierarchical kernel descriptors [30]. Note that, our RandLA-Net can be wider and deeper by altering the number of layers, feature channels, and adjusting the sampling rate. However, larger models require additional computation and may more easily lead to overfitting. The details of each component are as follows:

Network Input: The input is a large-scale point cloud with a size of $\tilde{N} \times d_{in}$ (the batch dimension is dropped for simplicity), where $\tilde{N}$ is the total number of input points, d_{in} is the feature dimension of each input point. For both S3DIS [26] and Semantic3D [76] datasets, each point is represented by its 3D coordinates and color information (*i.e.*, x-y-z-R-G-B), while each point of the SemanticKITTI [28] dataset is only represented by 3D coordinates.

Encoding Layers: Four encoding layers are used in our network to progressively reduce the size of the point clouds and increase the per-point feature dimensions. Each encoding layer consists of a local feature aggregation module (Section 4.3.2) and a random sampling operation (Section 4.3.1). The point cloud is downsampled with a four-fold decimation ratio. In particular, only 25% of the point features are retained after each layer, *i.e.*, $(\tilde{N} \rightarrow \frac{\tilde{N}}{4} \rightarrow \frac{\tilde{N}}{16} \rightarrow \frac{\tilde{N}}{64} \rightarrow \frac{\tilde{N}}{256})$. Meanwhile, the

per-point feature dimension is gradually increased each layer to preserve more information, *i.e.*, $(8 \rightarrow 32 \rightarrow 128 \rightarrow 256 \rightarrow 512)$.

Decoding Layers: Four decoding layers are used after the encoding layers. For each layer in the decoder, we adopt the nearest-neighbor interpolation for efficiency and simplicity. In particular, the coordinates of all down-sampled points in each encoding layer are temporally stored for reference. For each query point in the decoding layers, we use the KNN algorithm to find the nearest neighboring point from the points of the previous layer. The features of the nearest point are copied to the target point. Subsequently, the upsampled feature maps are concatenated with the intermediate feature maps produced by encoding layers through skip connections, after which a shared MLP is applied to the concatenated feature vectors.

Final Semantic Prediction: The final semantic label of each point is obtained through three shared fully-connected layers $(\tilde{N}, 64) \rightarrow (\tilde{N}, 32) \rightarrow (\tilde{N}, n_{class})$ and a dropout layer. The dropout ratio is 0.5.

Network Output: The output of RandLA-Net is the predicted semantics of all points, with a size of $\tilde{N} \times n_{class}$, where n_{class} is the number of classes.

4.3.4 Implementation

We use the same network architecture for all the five large-scale open datasets, Semantic3D [76], SemanticKITTI [28], Toronto-3D [185], NPM3D [167], S3DIS [26]. The Adam optimizer [100] with default parameters is applied. The initial learning rate is set to 0.01 and decreases by 5% after each epoch. The network is trained for 100 epochs. We implement the KNN search in our framework based on the nanoflann⁶ package, which leverages an efficient KD-Tree data structure for fast search and query. In addition, we also use OpenMP for better parallelization, and the number of nearest points K is set to 16. For several extremely large datasets such as Semantic3D [76], which have more than 10^6 or even 10^8 points in a single point cloud, we crop sub-clouds to feed into our RandLA-Net. During training, we sample a fixed number of points ($\sim 10^5$) from each point cloud as the input for

⁶<https://github.com/jlblancoc/nanoflann>

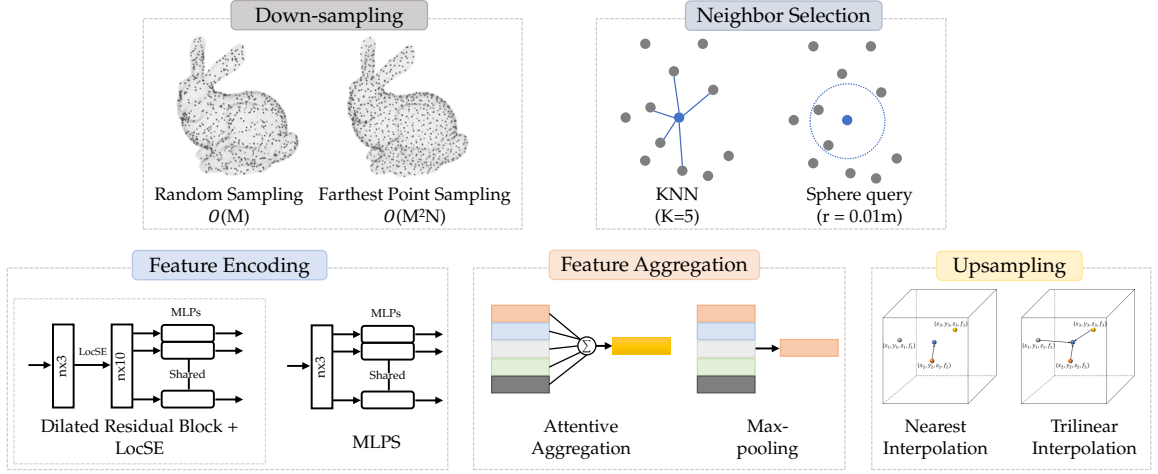


Figure 4.6: Comparative analysis at module and operation level between our proposed RandLA-Net and PointNet++. A visual representation is provided to highlight the intuitive differences between the two methods. Left: Our RandLA-Net, Right: PointNet++ [157].

parallelization. During testing, we iteratively infer several sub-clouds with overlaps to eventually cover all 3D points. Since many points have been inferred more than once, we follow [193] to use a simple voting scheme for better performance. To alleviate the problem of class imbalance, we use weighted cross-entropy as the loss function. In particular, the weight of each class is determined by its inverse frequency in the training split. We do not use any explicit data augmentation techniques such as rotation, scaling, and translation during training, since we have not found significant performance differences in practice. Note that, our random downsampling in each encoding layer can be regarded as implicit data augmentation, since even if the exact same inputs are used, the points retained after several random downsamplings are inevitably different. This can be viewed as a way to improve the diversity of the data to some extent, hence improving the generalization capacity of the proposed network through training. All experiments are conducted on an NVIDIA RTX2080Ti GPU.

4.3.5 Relation to Previous PointNet++

Here we discuss the relation between our RandLA-Net and the previous PointNet++ [157]. Although the two approaches share similar design philosophies in aspects such as local feature aggregation, the proposed RandLA-Net differs from PointNet++ in the following aspects:

- **Down-sampling.** To encode the input point clouds, PointNet++ uses computationally expensive, iterative farthest point sampling (FPS) to down-sample the point features. By contrast, a simple random sampling approach is used in our RandLA-Net, allowing the framework to be highly efficient and scalable for large-scale point clouds.
- **Neighbor Selection.** To learn local features, PointNet++ selects neighboring points within a sphere. It is thus able to maintain a geometrically consistent neighborhood, but the variable number of neighboring points within the sphere requires additional computation to be converted to a predefined and fixed number of points. By comparison, a simple and efficient K-Nearest Neighbours (KNN) is used in RandLA-Net.
- **Feature Encoding.** PointNet++ learns per-point features by using shared MLPs to encode the vanilla coordinates. Instead, we *explicitly* encode the coordinates, relative positions, and the relative distances of neighboring points through the LocSE unit, enabling better capture of local geometrical patterns. In addition, the dilated residual block is used to further increase the size of the receptive field for each individual point.
- **Feature Aggregation.** PointNet++ aggregates neighboring feature sets using the simple max-pooling operation, resulting in the majority of neighboring point information being lost. In contrast, our RandLA-Net uses attentive pooling to automatically select important features in a data-driven manner, hence the prominent features are more likely to be preserved.
- **Up-sampling.** PointNet++ uses trilinear interpolation to upsample the point features in the decoder, which introduces considerable extra computation due to poor parallelization. Our RandLA-Net simply duplicates the features of the nearest neighboring point, which does not require additional computation, therefore being more efficient than that of PointNet++.

We also provide an intuitive comparison of these two methods in Figure 4.6 for clarity.

4.4 Experiments

We first evaluate the efficiency of existing sampling approaches in Section 4.4.1, and compare the efficiency of our RandLA-Net with existing networks in Section 4.4.2. Secondly, we conduct extensive experiments on multiple benchmarks to compare the semantic segmentation in Section 4.4.3. At last, our ablation study is presented in Section 4.4.4.

4.4.1 Efficiency of Random Sampling

In this section, we empirically evaluate the efficiency of existing sampling approaches including FPS, IDIS, PDS, RS, GS, CRS, and PGS, which have been discussed in Section 4.3.1. In particular, we conduct the following 4 groups of experiments.

- Group 1. Given a small-scale point cloud ($\sim 10^3$ points), we use each sampling approach to progressively downsample it. Specifically, the point cloud is downsampled by five steps with only 25% points being retained in each step on a single GPU i.e. a four-fold decimation ratio. This means that there are only $\sim (1/4)^5 \times 10^3$ points left in the end. This downsampling strategy emulates the procedure used in PointNet++ [157]. For each sampling approach, we sum up its time and memory consumption for comparison.
- Group 2/3/4. The total number of points is increased towards large-scale, *i.e.*, around 10^4 , 10^5 and 10^6 points respectively. We use the same five sampling steps as in Group 1.

Analysis. Figure 4.7 compares the time and memory consumption of each sampling approach to process different scales of point clouds. It can be seen that: 1) For small-scale point clouds ($\sim 10^3$), all sampling approaches tend to have similar time and memory consumption, and are unlikely to incur a heavy or limiting computation burden. 2) For large-scale point clouds ($\sim 10^6$), FPS/IDIS/PDS/GS/CRS/PGS are either extremely time-consuming or memory-costly for large-scale computation. By contrast, random sampling has superior time and memory efficiency overall. This result clearly demonstrates that most existing networks [157, 122, 222, 131, 255, 240] are only able to be optimized on small blocks of point clouds primarily because they rely on the expensive sampling approaches. Motivated by this, we use the efficient random sampling strategy in our RandLA-Net.

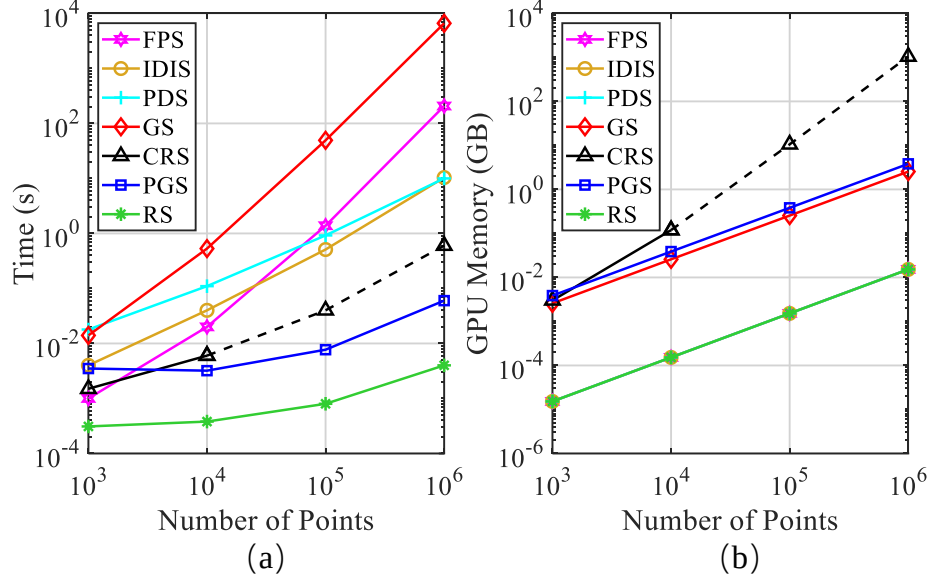


Figure 4.7: Time and memory consumption of different sampling approaches. The dashed lines represent estimated values due to the limited GPU memory. Note that, the curves of GPU memory consumption for RPS, PDS, FPS, and IDIS overlap together, because these sampling methods mainly run on the CPU.

4.4.2 Efficiency of RandLA-Net

In this section, we systematically evaluate the overall efficiency of our RandLA-Net on real-world large-scale point clouds for semantic segmentation. Particularly, we evaluate RandLA-Net on the SemanticKITTI [28] dataset, showing the detailed runtime of each component. We also evaluate the time consumption of recent representative works [155, 157, 110] on the same dataset. For a fair comparison, we feed the same number of points (*i.e.*, 12×81920 , the batch size is 12) from the sequence into each neural network.

In addition, we also evaluate the memory consumption of RandLA-Net and the baselines. In particular, we report the peak memory (GPU) of each network with the same input point clouds during inference. Note that, all experiments are conducted on the same machine with an AMD 3700X @3.6GHz CPU and an NVIDIA RTX2080Ti GPU.

Analysis. Table 4.2 quantitatively shows the total time and memory consumption of different approaches. It can be seen that, 1) SPG [110] has the lowest GPU memory consumption, but takes the longest time to process the point clouds due to the expensive geometrical partitioning and super-graph construction steps; 2) PointNet++ [157] is also computationally expensive mainly because of the FPS sampling operation; 3) PointNet [155] are unable to take extremely large-scale

Table 4.2: The run-time decomposition and peak memory of different approaches for semantic segmentation of input point clouds with 12×81920 ($\sim 10^6$) points, which is sampled from the LiDAR point clouds on Sequence 08 (around 80K-120K points per frame) of the SemanticKITTI [28] dataset. In particular, [†]this is achieved by inference half of the input point clouds twice, since PointNet is unable to process the whole inputs in a single pass. [‡]The original implementation of PointNet++ adopts an extremely inefficient KNN during trilinear interpolation. Hence, we also report the result of re-implemented PointNet++^{*} for a fair comparison. Note that, because SPG does not have explicit sampling and neighboring search modules, its run-time decomposition is not reported. We measure the peak memory consumption based on the [TensorFlow profiler](#). Note the peak memory may be different from the actual allocated GPU memory.

	Time (s)				Peak Memory
	Sampling	Neighbour.	Network.	Total	
PointNet (Vanilla) [†] [155]	0	0	0.46 [†]	0.46	9064Mb
PointNet++ (SSG) [‡] [157]	1.32	0.32+26.98 [‡]	0.20	28.82	4135Mb
PointNet++ (re-imp.) [*] [157]	1.32	0.52	0.20	2.04	4135Mb
SPG [110]	-	-	-	99.67	1663Mb
RandLA-Net (Ours)	0.01	0.40	0.15	0.55	5198Mb

point clouds (*e.g.*, 10^6 points) in a single pass due to their memory inefficient operations. 4) Thanks to the simple random sampling together with the efficient MLP-based local feature aggregator, our RandLA-Net takes around 0.5s to infer the semantic labels of large-scale point clouds with nearly 1 million points (12×81920 points).

4.4.3 Semantic Segmentation on Benchmarks

In this section, we evaluate the semantic segmentation of our RandLA-Net on multiple large-scale public datasets: the outdoor Semantic3D [76], SemanticKITTI [28], Toronto-3D [185], NPM3D [167], DALES [200], and the indoor S3DIS [26], ScanNet [55]. For a fair comparison, we follow KPConv [193] to pre-process the whole input point clouds by using the grid-sampling strategy at the beginning. In fact, this is likely to reduce the point density and help increase the actual receptive field for each 3D point.

4.4.3.1 Evaluation on Semantic3D

The Semantic3D dataset [76] consists of 15 point clouds for training and 15 for online testing. Each point cloud has up to 10^8 points, covering up to $160 \times 240 \times 30$ meters in real-world 3D space. The raw 3D points belong to 8 classes and contain

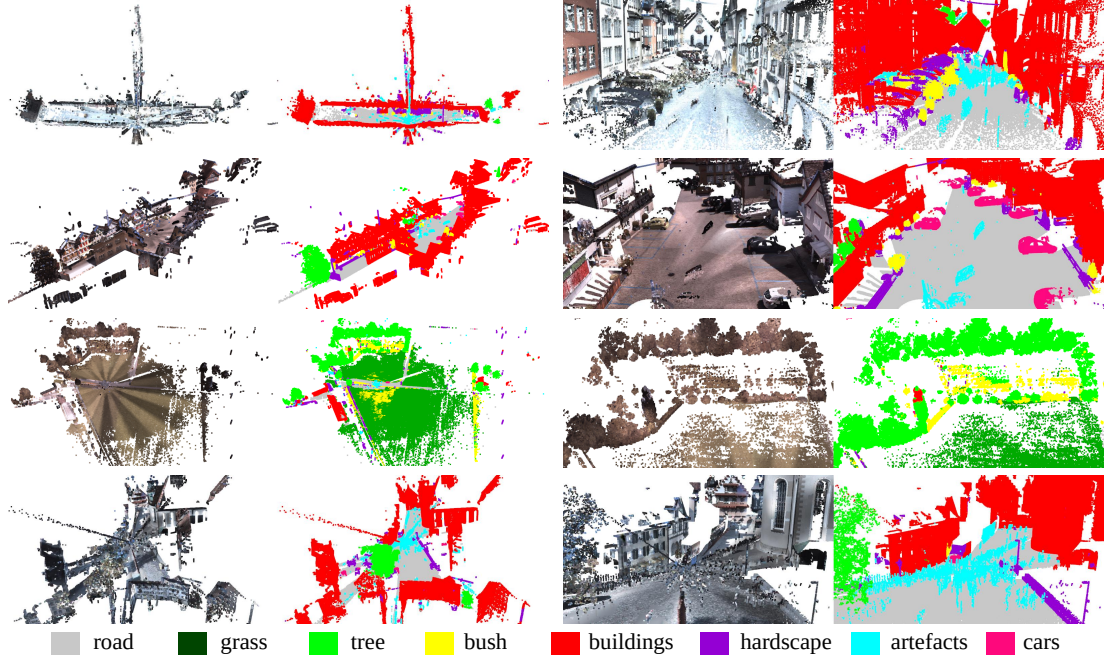


Figure 4.8: Qualitative results of RandLA-Net on the *reduced-8* split of Semantic3D. From left to right: full RGB colored point clouds, predicted semantic labels of full point clouds, detailed view of colored point clouds, detailed view of predicted semantic labels. Note that the ground truth of the test set is not publicly available.

3D coordinates, RGB information, and intensity. We only use the 3D coordinates and color information to train and test our RandLA-Net. Overall Accuracy (OA) and mean Intersection-over-Union (mIoU) of all classes are used as the evaluation metrics. For a fair comparison, we only include the results of recently published strong baselines [32, 190, 192, 166, 254, 208, 110, 193] and the current state-of-the-art approach RGNet [195].

Table 4.3 presents the quantitative results of different approaches achieved on the *reduced-8* test set. RandLA-Net clearly outperforms all existing methods in terms of both mIoU and OA. Notably, RandLA-Net also achieves superior performance on six of the eight classes, except *low vegetation* and *scanning artefact*. We also report the results on *semantic-8* subset on Table 4.4. Note that, compared with the *reduced-8* subset, the *semantic-8* subset has $10\times$ more points and much stronger category imbalance, and therefore it is more challenging. Our RandLA-Net also achieves the best performance in terms of overall accuracy, but ranks third in mIoU performance, primary because ConvPoint [31] and SPGraph [110] achieves much better results on *hard scape* and *scanning artifact*. Figure 4.8 shows the qualitative results of RandLA-Net on this dataset.

Table 4.3: Quantitative results of different approaches on Semantic3D (*reduced-8*) [76]. This test consists of 78,699,329 points. The scores are obtained from the recent publications.

Methods	mIoU(%)	OA(%)	man-made.	natural.	high veg.	low veg.	buildings	hard scape	scanning art.	cars
SnapNet. [32]	59.1	88.6	82.0	77.3	79.7	22.9	91.1	18.4	37.3	64.4
SEGCloud [190]	61.3	88.1	83.9	66.0	86.0	40.5	91.1	30.9	27.5	64.3
RF.MSSF [192]	62.7	90.3	87.6	80.3	81.8	36.4	92.2	24.1	42.6	56.6
MSDeepVoxNet [166]	65.3	88.4	83.0	67.2	83.8	36.7	92.4	31.3	50.0	78.2
ShellNet [254]	69.3	93.2	96.3	90.4	83.9	41.0	94.2	34.7	43.9	70.2
GACNet [208]	70.8	91.9	86.4	77.7	88.5	60.6	94.2	37.3	43.5	77.8
SPG [110]	73.2	94.0	97.4	92.6	87.9	44.0	83.2	31.0	63.5	76.2
KPConv [193]	74.6	92.9	90.9	82.2	84.2	47.9	94.9	40.0	77.3	79.7
RGNNet [195]	74.7	94.5	97.5	93.0	88.1	48.1	94.6	36.2	72.0	68.0
RandLA-Net (Ours)	77.4	94.8	95.6	91.4	86.6	51.5	95.7	51.5	69.8	76.8

Table 4.4: Quantitative results of different approaches on Semantic3D (*semantic-8*) [76]. This test consists of 2,091,952,018 points. The scores are obtained from the recent publications.

Methods	mIoU(%)	OA(%)	man-made.	natural.	high veg.	low veg.	buildings	hard scape	scanning art.	cars
TML-PC [147]	39.1	74.5	80.4	66.1	42.3	41.2	64.7	12.4	0.	5.8
TMLC-MS [77]	49.4	85.0	91.1	69.5	32.8	21.6	87.6	25.9	11.3	55.3
PointNet++ [157]	63.1	85.7	81.9	78.1	64.3	51.7	75.9	36.4	43.7	72.6
EdgeConv [53]	64.4	89.6	91.1	69.5	65.0	56.0	89.7	30.0	43.8	69.7
SnapNet [32]	67.4	91.0	89.6	79.5	74.8	56.1	90.9	36.5	34.3	77.2
PointGCR [139]	69.5	92.1	93.8	80.0	64.4	66.4	93.2	39.2	34.3	85.3
RGNNet [195]	72.0	90.6	86.4	70.3	69.5	68.0	96.9	43.4	52.3	89.5
LCP [33]	74.6	94.1	94.7	85.2	77.4	70.4	94.0	52.9	29.4	92.6
SPGraph [110]	76.2	92.9	91.5	75.6	78.3	71.7	94.4	56.8	52.9	88.4
ConvPoint [31]	76.5	93.4	92.1	80.6	76.0	71.9	95.6	47.3	61.1	87.7
RandLA-Net (Ours)	75.8	95.0	97.4	93.0	70.2	65.2	94.4	49.0	44.7	92.7

4.4.3.2 Evaluation on SemanticKITTI

SemanticKITTI [28] consists of 43552 densely annotated LiDAR scans belonging to 22 sequences. Each scan is a large-scale point cloud with $\sim 10^5$ points and spanning up to $160 \times 160 \times 20$ meters in 3D space. Officially, the sequences 00~07 and 09~10 (19130 scans) are used for training, the sequence 08 (4071 scans) for validation, and the sequences 11~21 (20351 scans) for online testing. The raw 3D points only have 3D coordinates without color information. The mIoU score over 19 categories is used as the standard metric.

Table 4.5 shows a quantitative comparison of our RandLA-Net with two families of recent approaches, *i.e.*, 1) point-based methods [155, 110, 179, 157, 189] and 2) projection based approaches [218, 219, 28, 144, 163], and Figure 4.9 shows some qualitative results of RandLA-Net on the validation split. It can be seen that our RandLA-Net surpasses all point based approaches [155, 110, 179, 157, 189] by large margins, with remarkable improvement of 15% compared with the second-best approach. We also outperform most projection based methods [218, 219, 28], but not significantly, primarily because SqueezeSegV3 [230] achieves much better re-

Table 4.5: Quantitative results of different approaches on SemanticKITTI [28]. The scores are obtained from the recent publications.

Methods	Size	mIoU(%)	Params(M)	road	sidewalk	parking	other-ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic-sign
PointNet [155]	50K pts	14.6	3	61.6	35.7	15.8	1.4	41.4	46.3	0.1	1.3	0.3	0.8	31.0	4.6	17.6	0.2	0.2	0.0	12.9	2.4	3.7
SPG [110]		17.4	0.25	45.0	28.5	0.6	0.6	64.3	49.3	0.1	0.2	0.2	0.8	48.9	27.2	24.6	0.3	2.7	0.1	20.8	15.9	0.8
SPLATNet [179]		18.4	0.8	64.6	39.1	0.4	0.0	58.3	58.2	0.0	0.0	0.0	0.0	71.1	9.9	19.3	0.0	0.0	0.0	23.1	5.6	0.0
PointNet++ [157]		20.1	6	72.0	41.8	18.7	5.6	62.3	53.7	0.9	1.9	0.2	0.2	46.5	13.8	30.0	0.9	1.0	0.0	16.9	6.0	8.9
TangentConv [189]		40.9	0.4	83.9	63.9	33.4	15.4	83.4	90.8	15.2	2.7	16.5	12.1	79.5	49.3	58.1	23.0	28.4	8.1	49.0	35.8	28.5
SqueezeSeg [218]	64*2048 pixels	29.5	1	85.4	54.3	26.9	4.5	57.4	68.8	3.3	16.0	4.1	3.6	60.0	24.3	53.7	12.9	13.1	0.9	29.0	17.5	24.5
SqueezeSegV2 [219]		39.7	1	88.6	67.6	45.8	17.7	73.7	81.8	13.4	18.5	17.9	14.0	71.8	35.8	60.2	20.1	25.1	3.9	41.1	20.2	36.3
DarkNet21Seg [28]		47.4	25	91.4	74.0	57.0	26.4	81.9	85.4	18.6	26.2	26.5	15.6	77.6	48.4	63.6	31.8	33.6	4.0	52.3	36.0	50.0
DarkNet53Seg [28]		49.9	50	91.8	74.6	64.8	27.9	84.1	86.4	25.5	24.5	32.7	22.6	78.3	50.1	64.0	36.2	33.6	4.7	55.0	38.9	52.2
RangeNet53++ [144]		52.2	50	91.8	75.2	65.0	27.8	87.4	91.4	25.7	34.4	23.0	80.5	55.1	64.6	38.3	38.8	4.8	58.6	47.9	55.9	
SalsaNext [54]		54.5	6.73	90.9	74.0	58.1	27.8	87.9	90.9	21.7	36.4	29.5	19.9	81.8	61.7	66.3	52.0	52.7	16.0	58.2	51.7	58.0
SqueezeSegV3 [230]		55.9	26	91.7	74.8	63.4	26.4	89.0	92.5	29.6	38.7	36.5	33.0	82.0	58.7	65.4	45.6	46.2	20.1	59.4	49.6	58.9
LatticeNet [163]		52.2	-	88.8	73.8	64.6	25.6	86.9	88.6	43.3	12.0	20.8	24.8	76.4	57.9	54.7	34.2	39.9	60.9	55.2	41.5	42.7
PolarNet [250]	-	54.3	14	90.8	74.4	61.7	21.7	90.0	93.8	22.9	40.2	30.1	28.5	84.0	65.5	67.8	43.2	40.2	5.6	61.3	51.8	57.5
RandLA-Net (Ours)	50K pts	55.9	1.24	90.5	74.0	61.8	24.5	89.7	94.2	43.9	47.4	32.2	39.1	83.8	63.6	68.6	48.4	47.4	9.4	60.4	51.0	50.7

sults on the small object category such as *motorcyclist*. However, our RandLA-Net is more lightweight with fewer parameters compared with the projection based methods.

Table 4.6: Quantitative results of different approaches on Toronto3D [185]. The scores of the baselines are obtained from [185].

Methods	OA(%)	mIoU(%)	Road	Rd mrk.	Natural	Building	Util. line	Pole	Car	Fence
PointNet++ [157]	84.88	41.81	89.27	0.00	69.06	54.16	43.78	23.30	52.00	2.95
PointNet++ (MSG) [157]	92.56	59.47	92.90	0.00	86.13	82.15	60.96	62.81	76.41	14.43
DGCNN [213]	94.24	61.79	93.88	0.00	91.25	80.39	62.40	62.32	88.26	15.81
KPFCNN [193]	95.39	69.11	94.62	0.06	96.07	91.51	87.68	81.56	85.66	15.72
MS-PCNN [138]	90.03	65.89	93.84	3.83	93.46	82.59	67.80	71.95	91.12	22.50
TGNet [123]	94.08	61.34	93.54	0.00	90.83	81.57	65.26	62.98	88.73	7.85
MS-TGNet [185]	95.71	70.50	94.41	17.19	95.72	88.83	76.01	73.97	94.24	23.64
RandLA-Net (Ours, w/o RGB)	92.95	77.71	94.61	42.62	96.89	93.01	86.51	78.07	92.85	37.12
RandLA-Net (Ours, w/ RGB)[†]	94.37	81.77	96.69	64.21	96.92	94.24	88.06	77.84	93.37	42.86

4.4.3.3 Evaluation on Toronto-3D

The Toronto-3D dataset [185] consists of 78.3 million points acquired by the mobile laser systems (MLS), which covers approximately 1KM of urban outdoor 3D space. Following [185], the section *L002* is used for testing and the remaining three sections are used for training. OA and mIoU over 8 categories are used as evaluation metrics.

Table 4.6 reports the quantitative results achieved by our RandLA-Net with several recently published baselines [157, 213, 193, 138, 123, 185], and Figure 4.11 shows the qualitative results. Note that, only the 3D coordinates of each point are feed into the network for fair comparison [185]. It can be seen that our RandLA-Net

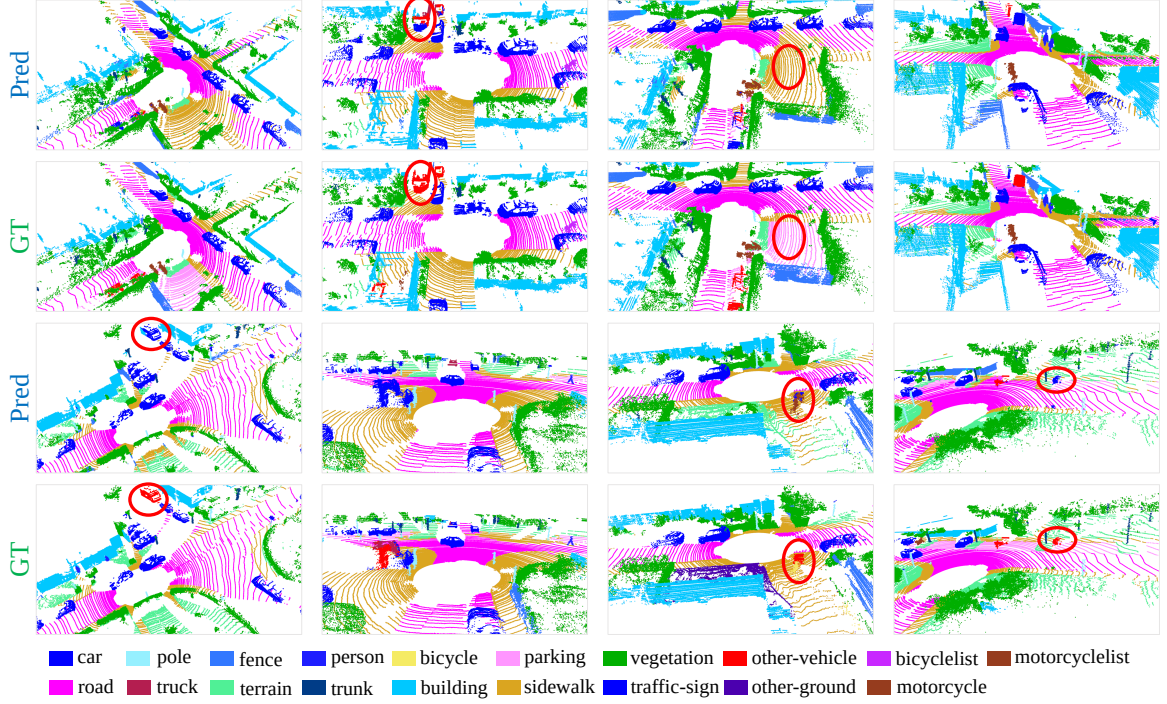


Figure 4.9: Qualitative results of RandLA-Net on the validation set of SemanticKITTI [28]. Red boxes show the failure cases.

achieves the best results on this dataset in terms of mIoU, improving the previous state-of-the-art MS-TGNet [185] from 70.50% to 77.71%.

It is observed that *Road marking* is the hardest to be recognized by all approaches. Primarily, this is because the pure geometric patterns of *Road marking* can not be easily distinguished against its neighbourhood. Particularly, given only 3D coordinates of points, it is virtually impossible to differentiate the *Road marking* and *Road*, as illustrated in Figure 4.11. This motivates us to include the appearance of points to further improve the accuracy. We simply train our RandLA-Net with both 3D coordinates and color information as input. As shown in the last row of

Table 4.7: Quantitative results of different approaches on NPM3D [167]. Mean IoU (mIoU, %), and per-class IoU (%) are reported. The scores are obtained from the recent publications.

Methods	mIoU(%)	Ground	Building	Pole	Bollard	Trash can	Barrier	Pedestrian	Car	Natural
RF_MSSF [192]	56.3	99.3	88.6	47.8	67.3	2.3	27.1	20.6	74.8	78.8
MS3_DVS [165]	66.9	99.0	94.8	52.4	38.1	36.0	49.3	52.6	91.3	88.6
HDGCN [124]	68.3	99.4	93.0	67.7	75.7	25.7	44.7	37.1	81.9	89.6
MS-RRFsegNet [136]	79.2	98.6	98.0	79.7	74.3	75.1	57.9	55.9	82.0	91.4
ConvPoint [31]	75.9	99.5	95.1	71.6	88.7	46.7	52.9	53.5	89.4	85.4
KPConv [193]	82.0	99.5	94.0	71.3	83.1	78.7	47.7	78.2	94.4	91.4
LCP [33]	82.7	99.6	98.1	77.2	91.1	64.7	66.5	58.1	95.6	93.9
RandLA-Net (Ours)	78.5	99.5	97.0	71.0	86.7	50.5	65.5	49.1	95.3	91.7

Table 4.8: Quantitative results of different approaches on S3DIS [26] (6-fold cross-validation). Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%) are reported.

Methods	OA(%)	mAcc(%)	mIoU(%)	ceil.	floor	wall	beam	col.	wind.	door	table	chair	sofa	book.	board	clut.
PointNet [155]	78.6	66.2	47.6	88.0	88.7	69.3	42.4	23.1	47.5	51.6	54.1	42.0	9.6	38.2	29.4	35.2
RSNet [94]	-	66.5	56.5	92.5	92.8	78.6	32.8	34.4	51.6	68.1	59.7	60.1	16.4	50.2	44.9	52.0
3P-RNN [241]	86.9	-	56.3	92.9	93.8	73.1	42.5	25.9	47.6	59.2	60.4	66.7	24.8	57.0	36.7	51.6
SPG [110]	86.4	73.0	62.1	89.9	95.1	76.4	62.8	47.1	55.3	68.4	73.5	69.2	63.2	45.9	8.7	52.9
PointCNN [122]	88.1	75.6	65.4	94.8	97.3	75.8	63.3	51.7	58.4	57.2	71.6	69.1	39.1	61.2	52.2	58.6
PointWeb [255]	87.3	76.2	66.7	93.5	94.2	80.8	52.4	41.3	64.9	68.1	71.4	67.1	50.3	62.7	62.2	58.5
ShellNet [254]	87.1	-	66.8	90.2	93.6	79.9	60.4	44.1	64.9	52.9	71.6	84.7	53.8	64.6	48.6	59.4
PointASNL [236]	88.8	79.0	68.7	95.3	97.9	81.9	47.0	48.0	67.3	70.5	71.3	77.8	50.7	60.4	63.0	62.8
KPConv. <i>rigid</i> [193]	-	78.1	69.6	93.7	92.0	82.5	62.5	49.5	65.7	77.3	57.8	64.0	68.8	71.7	60.1	59.6
KPConv. <i>deform</i> [193]	-	79.1	70.6	93.6	92.4	83.1	63.9	54.3	66.1	76.6	57.8	64.0	69.3	74.9	61.3	60.3
RandLA-Net (Ours)	88.0	82.0	70.0	93.1	96.1	80.6	62.4	48.0	64.4	69.4	69.4	76.4	60.0	64.2	65.9	60.1

Table 4.6, the IoU score of *Road marking* has been significantly improved by 22%, demonstrating the importance of appearance in this difficult case.

4.4.3.4 Evaluation on NPM3D

The NPM3D dataset [167] is a large-scale and high-quality urban point clouds dataset. Similar to the Toronto-3D dataset, the NPM3D dataset also consists of around 2KM MLS point cloud acquired in two cities. The mean IoU (mIoU) of the total 10 classes are used as the main evaluation metric.

Table 4.7 quantitatively shows the results achieved by our RandLA-Net and exiting baselines [31, 33, 193, 165, 192, 124]. In this dataset, only the 3D coordinates of each point is available. Due to the relatively lower scores of two minority classes *Trash can* and *Pedestrian*, our RandLA-Net achieves a mean IoU score of 78.5%, ranking after the latest LightConvPoint [33] and KPConv [193]. Primarily, the point features of *Trash can* and *Pedestrian* are very likely to be randomly dropped in our RandLA-Net because the two categories only have extremely small number of points scanned.

4.4.3.5 Evaluation on S3DIS

The S3DIS dataset [26] consists of 271 rooms belonging to 6 large areas. Each point cloud is a medium-sized single room ($\sim 20 \times 15 \times 5$ meters) with dense 3D points. Considering the relatively limited size of this dataset, we use 6-fold cross-validation to comprehensively evaluate the semantic segmentation performance of our RandLA-Net. As shown in Figure 4.10, we first divide the dataset into 6 subsets according to different areas, and then train the RandLA-Net model on selected 5 subsets and evaluated on the remaining 1 subset. This process is repeated

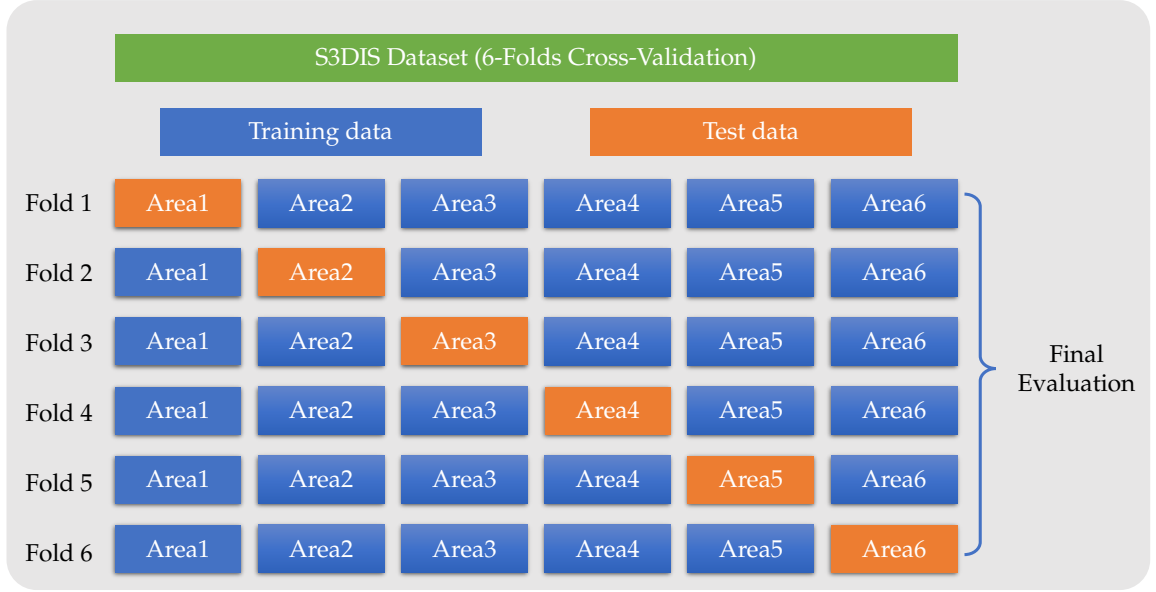


Figure 4.10: Illustration of the 6-fold cross-validation used in the S3DIS dataset.

6 times, with a different subset used for evaluation, respectively. The final segmentation results are the average results of different iterations. The mean IoU (mIoU), mean class Accuracy (mAcc) and Overall Accuracy (OA) of the total 13 classes are compared.

Table 4.8 quantitatively compares the performance of our RandLA-Net with existing baselines on this dataset. Our RandLA-Net achieves on-par or better performance than state-of-the-art methods. Note that, most of these baselines [157, 122, 255, 254, 213, 40] tend to use sophisticated but expensive operations or samplings to optimize the networks on small blocks (*e.g.*, 1×1 meter) of point clouds, and the relatively small rooms act in their favours to be divided into tiny blocks. By contrast, RandLA-Net is able to take the entire rooms as input and efficiently infer per-point semantics in a single pass.

Table 4.9: Quantitative results of different approaches on ScanNet (online test set).

Method	mIoU(%)	floor	wall	chair	sofa	table	door	cab	bed	desk	toil	sink	wind	pic	bkshf	curt	show	cntr	fridg	bath	other
ScanNet [55]	30.6	78.6	43.7	52.4	34.8	30.0	18.9	31.1	36.6	34.2	46.0	31.8	18.2	10.2	50.1	0.2	15.2	21.1	24.5	20.3	14.5
PointNet++ [157]	33.9	67.7	52.3	36.0	34.6	23.2	26.1	25.6	47.8	27.8	54.8	36.4	25.2	11.7	45.8	24.7	14.5	25.0	21.2	58.4	18.3
SPLATNET3D [179]	39.3	92.7	69.9	65.6	51.0	38.3	19.7	31.1	51.1	32.8	59.3	27.1	26.7	0.0	60.6	40.5	24.9	24.5	0.1	47.2	22.7
Tangent-Conv [189]	43.8	91.8	63.3	64.5	56.2	42.7	27.9	36.9	64.6	28.2	61.9	48.7	35.2	14.7	47.4	25.8	29.4	35.3	28.3	43.7	29.8
PointCNN [122]	45.8	94.4	70.9	71.5	54.5	45.6	31.9	32.1	61.1	32.8	75.5	48.4	47.5	16.4	35.6	37.6	22.9	29.9	21.6	57.7	28.5
PointConv [222]	55.6	94.4	76.2	73.9	63.9	50.5	44.5	47.2	64.0	41.8	82.7	54.0	51.5	18.5	57.4	43.3	57.5	43.0	46.4	63.6	37.2
SPH3D-GCN [117]	61.0	93.5	77.3	79.2	70.5	54.9	50.7	53.2	77.2	57.0	85.9	60.2	53.4	4.6	48.9	64.3	70.2	40.4	51.0	85.8	41.4
KPConv [193]	68.4	93.5	81.9	81.4	78.5	61.4	59.4	64.7	75.8	60.5	88.2	69.0	63.2	18.1	78.4	77.2	80.5	47.3	58.7	84.7	45.0
SparseConvNet [73]	72.5	95.5	86.5	86.9	82.3	62.8	61.4	72.1	82.1	60.3	93.4	72.4	68.3	32.5	84.6	75.4	87.0	53.3	71.0	64.7	57.2
SegGCN [115]	58.9	93.6	77.1	78.9	70.0	56.3	48.4	51.4	73.1	57.3	87.4	59.4	49.3	6.1	53.9	46.7	50.7	44.8	50.1	83.3	39.6
RandLA-Net (Ours)	64.5	94.5	79.2	82.9	73.8	59.9	52.3	57.7	73.1	47.7	82.7	61.8	62.1	26.9	69.9	73.6	74.9	44.6	48.4	77.8	45.4

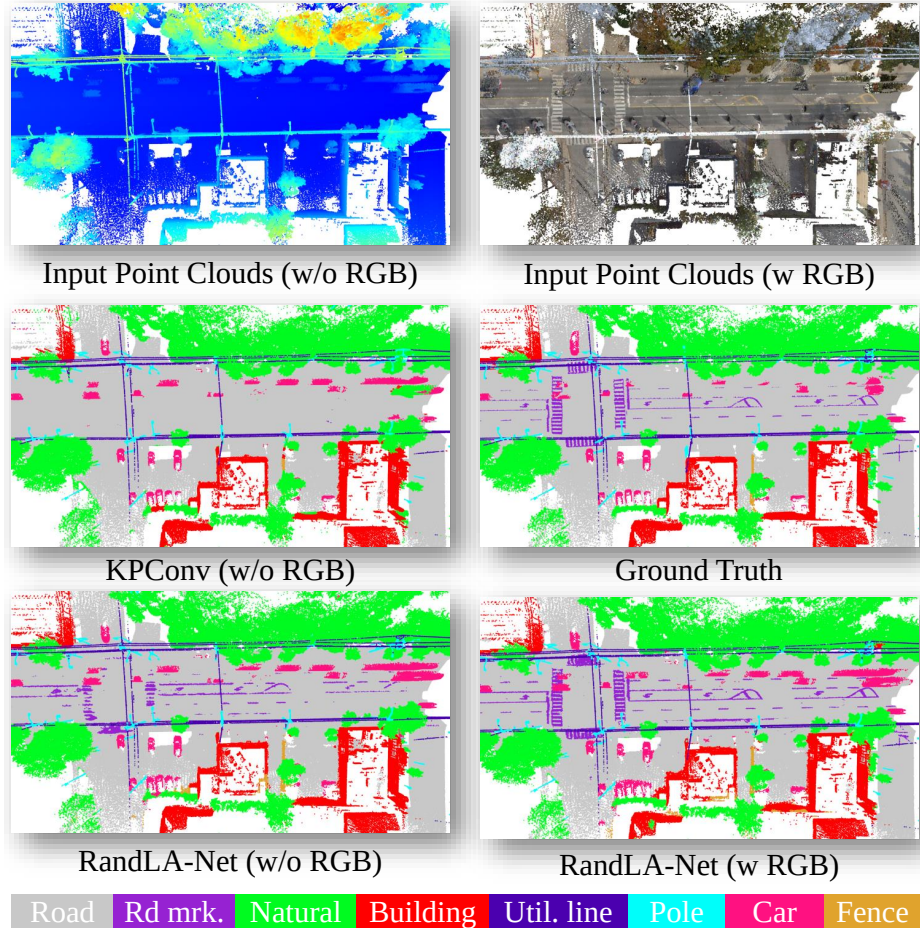


Figure 4.11: Semantic segmentation results of KPConv [193] and our approach on Toronto-3D [185]. Our method clearly performs better on the category of road marking.

4.4.3.6 Evaluation on ScanNet

We conduct an experiment on the ScanNet dataset to further evaluate the performance of our method. The detailed results on the online test set are shown in Table 4.9.

It can be seen that our approach achieves satisfactory results on this dataset, but not the best. Our mean IoU score is lower than KPConv [193] and SparseConvNet [73], because our RandLA-Net performs much worse on several small categories such as *refrigerator* and *cabinet*. One possible reason is that the major information of small objects may be lost due to the aggressive random sampling used in our framework. Note that, the ScanNet dataset is mainly composed of relatively small indoor scenes (average spatial size: $5\text{m} \times 5\text{m} \times 2\text{m}^7$), hence it is not in favor of

⁷This is computed by averaging the spatial sizes of all scenes in the dataset.

Table 4.10: Quantitative results of different approaches on the DALES dataset. Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%) are reported.

Method	OA(%)	mIoU(%)	<i>ground</i>	<i>buildings</i>	<i>cars</i>	<i>trucks</i>	<i>poles</i>	<i>power lines</i>	<i>fences</i>	<i>vegetation</i>
ShellNet [254]	96.4	57.4	96.0	95.4	32.2	39.6	20.0	27.4	60.0	88.4
PointCNN [122]	97.2	58.4	97.5	95.7	40.6	4.80	57.6	26.7	52.6	91.7
SuperPoint [110]	95.5	60.6	94.7	93.4	62.9	18.7	28.5	65.2	33.6	87.9
ConvPoint [31]	97.2	67.4	96.9	96.3	75.5	21.7	40.3	86.7	29.6	91.9
PointNet++ [157]	95.7	68.3	94.1	89.1	75.4	30.3	40.0	79.9	46.2	91.2
KPConv [193]	97.8	81.1	97.1	96.6	85.3	41.9	75.0	95.5	63.5	94.1
RandLA-Net (Ours)	97.1	80.0	97.0	93.2	83.7	43.8	59.4	94.8	71.5	96.6

our method which is primarily designed for large-scale point clouds (*e.g.*, point clouds spanning up to $250\text{m} \times 260\text{m} \times 80\text{m}$ in Semantic3D dataset). We observe that the performance gap between our approach and KPConv on the ScanNet is greater than that on the S3DIS dataset. This may be caused by the KNN used in our framework, where the receptive field of each point is not stable as that of the spherical neighborhood used in KPConv, especially for the RGBD-reconstructed dataset such as ScanNet (extremely non-uniform, incomplete, and noisy.)

4.4.3.7 Evaluation on DALES

The DALES dataset [200] consists of large-scale point clouds with 505 million labeled points acquired by the aerial laser systems (ALS), which covers urban areas spanning 10 km^2 . Each point is labeled with one of 8 semantic categories, and there is no color information available in this dataset.

Following the evaluation protocol in [200], we further compare the segmentation performance of our RandLA-Net with other approaches on this dataset. The detailed results are shown in Table 4.10. It can be seen that our RandLA-Net also achieves satisfactory overall performance on this dataset, with the mean IoU score of 80.0%. It also achieves the best performance on categories such as *trucks* and *vegetation*. We also notice that the performance is slightly lower than KPConv [193], primarily because of the poor performance on the *poles*. This is likely caused by the aggressive downsampling used in our framework.

4.4.3.8 Evaluation on SensatUrban

As mentioned in Chapter 3, SensatUrban is an Urban-scale photogrammetry point cloud dataset, including nearly 3 billion labeled points in 13 semantic categories. Here, we further evaluate the performance of the proposed RandLA-Net on this

Table 4.11: Evaluation of the proposed RandLA-Net [90] on SensatUrban [89] benchmark. We evaluate on 13 predefined semantic categories using the Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%).

	OA(%)	mAcc(%)	mIoU(%)	ground	veg.	building	wall	bridge	parking	rail	traffic.	street.	car	footpath	bike	water
PointNet [155]	80.78	30.32	23.71	67.96	89.52	80.05	0.00	0.00	3.95	0.00	31.55	0.00	35.14	0.00	0.00	0.00
PointNet++ [157]	84.30	39.97	32.92	72.46	94.24	84.77	2.72	2.09	25.79	0.00	31.54	11.42	38.84	7.12	0.00	56.93
TangentConv [189]	76.97	43.71	33.30	71.54	91.38	75.90	35.22	0.00	45.34	0.00	26.69	19.24	67.58	0.01	0.00	0.00
SPGraph [110]	85.27	44.39	37.29	69.93	94.55	88.87	32.83	12.58	15.77	15.48	30.63	22.96	56.42	0.54	0.00	44.24
SparseConv [73]	88.66	63.28	42.66	74.10	97.90	94.20	63.30	7.50	24.20	0.00	30.10	34.00	74.40	0.00	0.00	54.80
KPConv [193]	93.20	63.76	57.58	87.10	98.91	95.33	74.40	28.69	41.38	0.00	55.99	54.43	85.67	40.39	0.00	86.30
RandLA-Net [90]	91.24	74.68	58.14	82.23	98.39	92.69	56.62	49.00	54.19	25.10	60.98	38.69	83.42	38.74	0.00	75.80

dataset. The experimental procedures for training and testing in this section were kept consistent with those outlined in Section 3.4.1 for a fair comparison.

Table 4.11 shows that the proposed RandLA-Net has the highest mIoU performance on the SensatUrban dataset, slightly better than KPConv. It is also noted that RandLA-Net also has significantly higher mAcc scores than the other approaches, indicating a more balanced overall segmentation performance. This is further supported by the fact that RandLA-Net has the best segmentation performance on underrepresented categories such as *bridges* and *rail*, which are one of the most difficult categories in this dataset.

4.4.4 Ablation Studies

Table 4.12: The mean IoU scores of all ablated networks based on our full RandLA-Net.

Ablated networks	mIoU(%)
(1) Remove local spatial encoding	49.8
(2) Replace with max-pooling	55.2
(3) Replace with mean-pooling	53.4
(4) Replace with sum-pooling	54.3
(5) Simplify dilated residual block	48.8
(6) The Full framework (RandLA-Net)	57.1

Since the impact of random sampling is fully studied in Section 4.4.1, we conduct the following ablation studies for our local feature aggregation module. All ablated networks are trained on sequences 00~07 and 09~10, and tested on the sequence 08 of SemanticKITTI dataset [28].

Table 4.13: The mean IoU scores of our RandLA-Net with different designs of LocSE.

LocSE	mIoU(%)
(1) (p_i)	48.9
(2) (p_i^k)	50.7
(3) $(p_i - p_i^k)$	56.4
(4) $(\ p_i - p_i^k\)$	35.9
(5) (p_i, p_i^k)	52.5
(6) $(p_i, p_i^k, p_i - p_i^k)$	56.8
(7) $(p_i, p_i^k, \ p_i - p_i^k\)$	53.7
(8) $(p_i - p_i^k, \ p_i - p_i^k\)$	56.9
(9) $(p_i, p_i^k, p_i - p_i^k, \ p_i - p_i^k\)$ (The Full Unit)	57.1

4.4.4.1 Ablation of RandLA-Net Framework

Since our RandLA-Net stacks multiple of our key components: LocSE, Attentive Pooling and Dilated Residual Block, we therefore conduct the following 5 ablation studies to demonstrate the effectiveness of each component.

(1) *Removing local spatial encoding (LocSE).* This unit enables each 3D point to explicitly observe its local geometry. After removing locSE, we directly feed the local point features into the subsequent attentive pooling.

(2~4) *Replacing attentive pooling by max/mean/sum pooling.* The attentive pooling unit learns to automatically combine all local point features. By comparison, the widely used max/mean/sum poolings tend to hard select or combine features, therefore their performance may be sub-optimal.

(5) *Simplifying the dilated residual block.* The dilated residual block stacks multiple LocSE units and attentive poolings, substantially dilating the receptive field for each 3D point. By simplifying this block, we use only one LocSE unit and attentive pooling per layer, i.e. we do not chain multiple blocks as in our original RandLA-Net.

Table 4.12 compares the mIoU scores of all ablated networks. From this, we can see that: 1) The greatest impact is caused by the removal of the chained spatial embedding and attentive pooling blocks. This is highlighted in Figure 4.5, which shows how using two chained blocks allows information to be propagated from a wider neighbourhood, i.e. approximately K^2 points as opposed to just K . This is especially critical with random sampling, which is not guaranteed to preserve a particular set of points. 2) The removal of the local spatial encoding unit shows

the next greatest impact on performance, demonstrating that this module is necessary to effectively learn local and relative geometry context. 3) Removing the attention module diminishes performance by not being able to effectively retain useful features. From this ablation study, we can see how the proposed neural units complement each other to attain our state-of-the-art performance.

4.4.4.2 Ablation of LocSE

As designed in Section 4.3.2, our LocSE component encodes the relative point position based on the following equation:

$$\mathbf{r}_i^k = MLP\left(p_i \oplus p_i^k \oplus (p_i - p_i^k) \oplus \|p_i - p_i^k\|\right) \quad (4.9)$$

We further investigate the effects of different spatial information in our framework. Particularly, we conduct the following 8 ablative experiments for LocSE:

- (1-4) *Encoding the point p_i only, the neighboring points p_i^k only, the relative position $p_i - p_i^k$ only, and the Euclidean distance $\|p_i - p_i^k\|$ only, respectively.*
- (5) *Encoding the point p_i and its neighboring points p_i^k .*
- (6) *Encoding the point p_i , neighboring points p_i^k , and their relative position $p_i - p_i^k$.*
- (7) *Encoding the point p_i , neighboring points p_i^k , and their Euclidean distance $\|p_i - p_i^k\|$.*
- (8) *Encoding only the relative position $p_i - p_i^k$ and their Euclidean distance $\|p_i - p_i^k\|$.*

Table 4.13 compares the mIoU scores of our network with different designs of LocSE. We can see that: 1) Explicitly encoding all spatial information leads to the highest mIoU scores. 2) The relative position $p_i - p_i^k$ plays an important role in the LocSE component, primarily because the relative point position enables the network to be aware of the local geometric patterns. 3) Only encoding the point position p_i , p_i^k or Euclidean distance $\|p_i - p_i^k\|$ is sub-optimal, because the individual point positions and distance only are less informative for capturing the local geometric patterns.

4.4.4.3 Ablation of Dilated Residual Block

As shown in Figure 4.3, we stack two sets of LocSE and Attentive Pooling units as the standard dilated residual block to gradually increase the receptive field. To further evaluate how the number of aggregation units in our dilated residual block impact the entire network, we conduct the following two more ablative experiments.

(1) *Using only one set of LocSE and attentive pooling.* In this setting, the receptive field for each 3D point becomes smaller.

(2) *Using three sets of LocSE and attentive pooling.* The receptive field becomes exponentially larger.

Table 4.14: The mIoU scores of RandLA-Net using different number of aggregation units in each residual block.

Dilated residual block	mIoU(%)
(1) one set of LocSE and attentive pooling	52.9
(2) three sets of LocSE and attentive pooling	54.2
(3) two sets (The Standard Block)	57.1

Table 4.14 shows the mIoU scores of our RandLA-Net with different number of aggregation units in each dilated residual block. It can be seen that: 1) Only one set of LocSE and attentive pooling in the dilated residual block leads to a significant drop of the mIoU score, due to the limited receptive field. 2) Three sets of LocSE and attentive pooling do not improve the accuracy as expected. This is because the significantly increased receptive fields and the large number of trainable parameters tend to be overfitted.

4.4.5 Analysing Point Neighbours

In our RandLA-Net, we use the simple KNN to find a fixed number of neighboring points to extract local patterns for each 3D point. Intuitively, the choice of K in KNN may affect the overall performance. In addition, the spatial range of neighboring points searched by KNN varies if the point density changes, making it unable to maintain a geometrically consistent neighborhood for each point [192]. Alternatively, the neighboring points can be queried within a fixed spherical radius, guaranteeing the consistency of local boundaries. In this section, we dive deep into point neighbor search, evaluating the choice of K and comparing KNN with radius query.

4.4.5.1 Choice of K in KNN

Intuitively, the larger the K , the more complex the local geometric patterns that can be learned and the heavier computation required. We conduct 7 groups of experiments where K varies from 4 to 64. The evaluation setting and protocol are the same as used in Section 4.4.4.

Figure 4.12 (a) shows the variation of segmentation performance given different choices of K in KNN. It can be seen that, the network achieves a significant gain (mIoU score: 46% $\rightarrow$ 57%) when the number of neighbouring points K increases from 4 to 16, after which the mean IoU score drops gradually given a larger size of K . This demonstrates that, 1) a very small set of neighbouring points is not helpful for the network to learn useful point local features due to the limited receptive field; 2) an extremely large set of neighbouring points is also unlikely to improve the accuracy, since the learned complex local features tend to overfit the training examples and are not general.

4.4.5.2 KNN vs. Spherical Neighbour

To further investigate how the neighbouring query mechanism impacts the performance, we replace the KNN with spherical radius query in our RandLA-Net without modifying any other components. Similarly, we also conduct 7 groups of experiments by gradually increasing the radius of the first encoding layer from 0.1 to 0.7, and then doubling the radius after each sampling layer. For a fair comparison and better parallelization, we consistently select $K = 16$ neighborhood points in each spherical neighborhood. This is achieved by following PointNet++ [157] to downsample or pad within each neighboring point set.

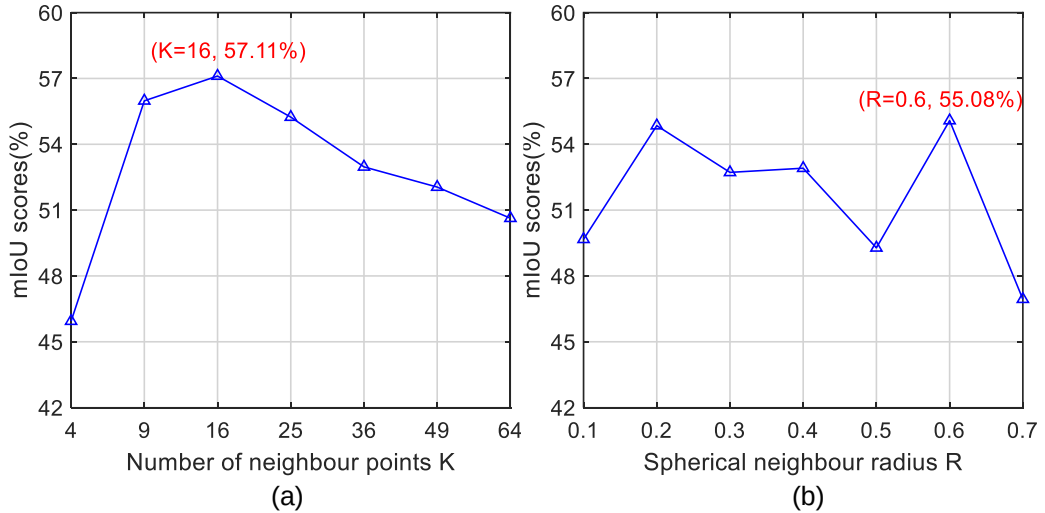


Figure 4.12: The results of our RandLA-Net with different neighbour searching methods. Left: The mIoU scores for different choices of K in KNN. Right: The mIoU scores for different choices of the radius R in spherical neighbours.

As shown in Fig 4.12 (b), the network achieves the best performance when the radius of the first layer is set to 0.6m, with a mean IoU score of 55.08%. This is

slightly lower than the best model with KNN (55.08% vs 57.11%), demonstrating that our framework is able to adapt to the spatial inconsistency potentially caused by KNN. Primarily, this is because the attentive pooling mechanism of our LocSE unit tends to automatically learn different weights for all neighbouring points, and the spatial locations of all points have been implicitly considered.

4.4.5.3 Attention Scores of Neighbouring Points

After the neighbouring points are selected by KNN, their features are aggregated by the attentive pooling as shown in Figure 4.3. To investigate how the features are effectively integrated, we visualize and analyze the learned attention scores in this section. As shown in Figure 4.13, for a specific center point, the learned attention scores for its 16 neighboring points tend to be similar in the first encoding layer, whereas the learned attention scores for its new 16 neighboring points in the subsequent layers are dramatically different. In effect, this is because the neighbouring points at the first layer are likely to be similar and equally informative, thus their learned attention scores are similar as well. After the large-scale point cloud is significantly downsampled and progressively aggregated, the retained point features tend to be highly compact and have different semantic context. Therefore, the attention scores of those neighbouring points which do not have similar semantic meanings with the center point, tend to be zero, while the most similar neighbouring point tends to have a high attention score.

4.4.6 Justification of Design Choices

4.4.6.1 Preprocessing

Different preprocessing steps are adopted for different segmentation approaches. For example, PointNet-based methods [155, 157, 222] adopt block random sampling (*e.g.*, sampling 4k points from 1m $\times$ 1m blocks) to preprocess the point clouds at the beginning, while KPConv [193] and SCN [73] first use grid sampling to process the point clouds. Here, we conduct additional experiments to further compare the segmentation performance of our method under different preprocessing steps. In particular, we evaluate the performance of our framework with grid sampling, block random sampling (1m $\times$ 1m with 4096 points and 6m $\times$ 6m with 40960 points). The detailed results are shown in Table 4.15.

It can be seen that our framework achieves comparable performance under different preprocessing steps when being fed with large point clouds (*i.e.*, grid

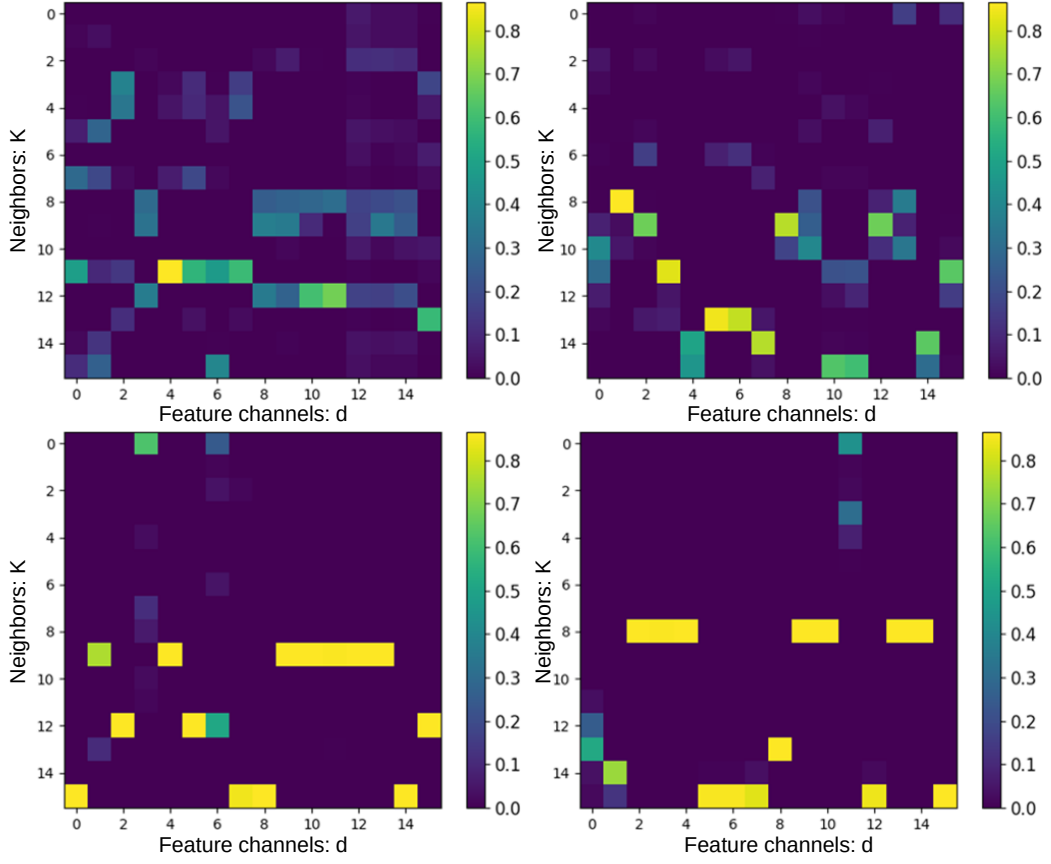


Figure 4.13: Visualization of the learned attention scores in different encoding layers. From top left to bottom right: the learned attention score map in the first (16×16), second (16×64), third (16×128), and the last encoding layers (16×256). For better visualization, we only keep the left part of the attention score map as an 16×16 image. The yellow color represents higher attention scores.

subsampling with 40960 points vs. block random sampling ($6m \times 6m$ with 40960 points)). However, the segmentation performance decreases from 61.63% to 59.10% when the block size is reduced to $1m \times 1m$, because the excessive block partitions severely break the geometrical structures and the network is hard to learn meaningful and robust representations. For the detailed analysis of the data preparation, please refer to [88].

Table 4.15: Quantitative results of the proposed RandLA-Net on the S3DIS dataset (Area 5) with different preprocessing steps.

Method	Size	mIoU(%)	ceil.	floor	wall	beam	col.	wind.	door	table	chair	sofa	book.	board	clut.
Grid (grid size=0.04m)	40960 pts	62.67	92.41	97.09	81.03	0.00	28.74	62.62	37.87	78.70	85.45	66.83	69.59	63.72	50.64
Block random ($1m \times 1m$)	4096 pts	59.10	90.27	96.53	77.82	0.00	30.32	65.28	36.10	72.02	79.82	55.00	64.48	56.30	44.33
Block random ($6m \times 6m$)	40960 pts	61.63	88.60	97.30	80.88	0.00	35.16	62.04	45.92	76.30	81.97	65.10	67.92	53.05	46.91

Table 4.16: Quantitative results of the proposed RandLA-Net on the SemanticKITTI dataset (validation set) with different upsampling strategies.

Methods	mIoU(%)
Random-upsampling	29.2
Trilinear-upsampling	53.0
Attentive upsampling (K=3)	56.6
Attentive upsampling (K=5)	57.0
Attentive upsampling (K=8)	56.1
Nearest-upsampling	57.1

4.4.6.2 Nearest-Neighbor Interpolation

As mentioned in Sec. 4.3.3, we choose the nearest-neighbor interpolation for simplicity and efficiency. However, our framework is flexible and allows the integration of different weighted interpolation strategies or other learnable layers. Therefore, we further compare the impact of different upsampling strategies on the overall segmentation performance, including random upsampling and weighted interpolation (*e.g.*, trilinear interpolation). In particular, we implement random upsampling by randomly copy the point features from the current feature map to create the upsampled feature map. For trilinear interpolation, we follow [157] to weight the point features of the nearest three points according to the Euclidean distance. The detailed results are shown in Table 4.16.

It can be seen that, (1) random upsampling is not applicable in the decoding stage, primarily because the skip connection used in our framework requires that the features at the same spatial location should be related, instead of being randomly selected. (2) the mIoU scores with trilinear interpolation and attentive upsampling are inferior to the nearest upsampling (57.1%). One possible reason is that the input point clouds are significantly down-sampled after all encoding layers, which means that the final subset of points preserved are very sparse and have different semantic meanings. Therefore, at the beginning stage of up-sampling, if multiple points with different semantic meanings are selected (*e.g.*, by KNN) to trilinearly interpolate the features of a new point, it might be detrimental. Instead, if only the nearest point is selected, it tends to be more robust.

4.4.6.3 Postprocessing: Voting Scheme

As presented in Section 4.3.4, the voting scheme is applied to obtain the final predictions during inference. To further investigate the impact of the voting scheme

on the overall segmentation performance, we conduct the following experiments by removing the voting scheme from our RandLA-Net. We obtained new mIoU scores on Toronto-3D [185], NPM3D [167], and SemanticKITTI [28] datasets. In particular, instead of voting from multiple runs, we only keep the per-point result in the last inference. The detailed results are shown in Table 4.17.

Table 4.17: The mIoU scores (%) of our RandLA-Net with and without the voting scheme.

	Toronto-3D [185] (Test set)	NPM3D [167] (Test set)	SemanticKITTI [28] (Validation set)
W/O voting	79.4	76.8	53.0
W/ voting	81.8	78.8	57.1

It can be seen that the overall mIoU scores of RandLA-Net has decreased by 2%-4% on Toronto-3D [185], NPM3D [167], and SemanticKITTI [28] datasets, showing that the commonly used voting scheme is helpful to boost the performance, yet it may incur extra computation burdens.

4.4.6.4 Sensitivity of RandLA-Net

Due to the usage of random sampling, the segmentation results of our RandLA-Net may be different given different runs. To further evaluate the sensitivity (*e.g.*, variance) of our method, we run the well trained model on the S3DIS dataset (Area 5) 5 times, with exactly the same experimental settings. The detailed results are shown in Table 4.18.

The average mIoU scores of our method on the Area5 of the S3DIS dataset is 62.44%, with a standard deviation of 0.75. We observe that the main difference of results lies in the minor categories such as *sofa*, *column*, *board* and *door*. Since these categories only have a small number of points, their segmentation results have noticeable fluctuations within multiple runs.

4.4.6.5 Variants of RandLA-Net

RandLA-Net can be deeper with more layers and wider with more channels. However, deeper or wider networks inevitably introduce larger models and massive training parameters, making the optimization more difficult in practice. Further, the overall computation efficiency would also be sacrificed.

In this section, we compare the segmentation performance of RandLA-Net with deeper layers and wider channels. In particular, for a deeper model, we increase

Table 4.18: Quantitative results of the proposed RandLA-Net on the S3DIS dataset (Area 5) with 5 runs. Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%) are reported.

	mIoU(%)	ceiling	floor	wall	beam	col.	wind.	door	table	chair	sofa	book.	board	clut.
Iter1	63.75	92.19	97.67	81.12	0.00	20.22	61.02	41.49	78.53	88.04	74.21	70.65	70.65	53.01
Iter2	61.56	92.51	96.18	80.92	0.00	20.00	60.63	37.93	76.37	87.68	58.63	70.33	67.13	51.94
Iter3	61.94	92.05	97.24	80.95	0.00	13.81	61.62	36.38	77.79	87.08	65.81	71.02	68.41	53.06
Iter4	62.30	92.28	97.59	80.37	0.00	23.11	60.99	33.32	80.01	87.42	63.46	70.42	68.45	52.45
Iter5	62.67	92.41	97.09	81.03	0.00	28.74	62.62	37.87	78.70	85.45	66.83	69.59	63.72	50.64
Average	62.44	92.29	97.15	80.88	0.00	21.18	61.38	37.40	78.28	87.13	65.79	70.40	67.67	52.22
STD	0.75	0.16	0.53	0.26	0.00	4.85	0.70	2.64	1.19	0.90	5.07	0.47	2.28	0.89

Table 4.19: Quantitative results of the proposed RandLA-Net on the SemanticKITTI dataset (validation set) with deeper layers and wider channels.

Methods	Params(M)	mIoU(%)	road	sidewalk	parking	other-ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic-sign
RandLA-Net (Deeper)	20.0	56.3	93.4	20.0	34.2	84.0	41.1	50.9	64.9	0.00	92.7	42.3	79.7	0.09	89.3	55.4	86.8	63.5	76.6	55.5	38.9
RandLA-Net (Wider)	4.9	57.2	94.8	19.9	35.2	68.6	48.0	56.5	71.2	0.00	93.5	45.3	80.8	1.80	89.7	55.9	86.7	66.0	74.3	57.8	41.1
RandLA-Net (Standard)	1.2	57.1	94.2	16.7	42.9	71.9	40.3	57.3	71.5	0.00	92.7	42.4	80.0	3.25	89.3	56.0	86.6	65.5	74.6	57.5	40.7

the number of encoding layers from 4 to 8, and the feature dimensions gradually increase from $(8 \rightarrow 16 \rightarrow 32 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 1024)$. Accordingly, the downsampling rate is decreased from 4 to 2 to preserve sufficient information for the decoder. For a wider model, we double the predefined per-point feature dimensions in our framework, that is: $(16 \rightarrow 64 \rightarrow 256 \rightarrow 512 \rightarrow 1024)$. All other hyperparameters such as the number of nearest neighbors, learning rate, batch size, and the number of points per batch remain unchanged for a fair comparison. The detailed results achieved on the SemanticKITTI dataset are shown in Table 4.19.

It can be seen that the segmentation performance is slightly improved when more channels are added (*i.e.*, the wider model), but slightly decreased when more layers are added (*i.e.*, the deeper model). One possible reason is that the deeper model has nearly 20 times more parameters, making it prone to overfitting and difficult to optimize. Additionally, the same hyperparameters may not always be optimal for each model.

4.4.6.6 Overfitting issue of RandLA-Net

Several point-based methods [193, 122, 222] tend to be overfitting training datasets. We present both the training and validation accuracy of our RandLA-Net in three datasets in Table 4.20.

Table 4.20: Training and validation accuracy of our RandLA-Net on three datasets.

Datasets	Overall Accuracy (%)	
	Training Set	Validation set
Toronto3D [185]	93.6	93.0
S3DIS (Area5) [26]	95.7	87.1
SemanticKITTI [28]	94.3	90.9

Overall, our RandLA-Net has good generalization ability, without suffering from overfitting. The reasons are two-fold. First, RandLA-Net is lightweight with far fewer network parameters (1.24M) than PointConv [222] (21.7M), KP-Conv [193] (14.3M), and PointCNN [122] (11.5M). These parameters are shared by a large amount of 3D points and their neighbours, thus capturing more general patterns. Second, the random sampling used in our framework can be regarded as an implicit way of data augmentation and regularization. With the same input point cloud, the preserved 3D points can be different in multiple training iterations, hence implicitly increasing the data diversity, thereby preventing the network from overfitting the training set.

4.5 Discussion

In this chapter, we presented RandLA-Net, a novel algorithm for 3D point cloud semantic segmentation that leverages the advantages of random sampling to improve computational and memory efficiency. We also introduce a dedicated local feature aggregation module that hierarchically increases the receptive field of each point, preserving local geometric patterns and improving segmentation performance. The experimental results demonstrate the effectiveness of the proposed RandLA-Net, which is up to 200 times faster than existing approaches on large-scale point clouds, making it a suitable choice for processing large-scale point clouds.

However, our algorithm also has some limitations. Firstly, the use of KNN for searching point neighbors does not scale linearly and can be a major computation bottleneck when processing extremely large point clouds. To address this issue, one potential solution is to use the point-voxel representation [134, 233] or K-random path selection from KD-Tree or OcTree. Secondly, the random sampling used in our framework only focuses on the observed points and may miss potentially important information in missing or occluded regions. Therefore, it would

be interesting to generate new random point sets to learn better local patterns. Finally, combining the efficiency of random sampling and the effectiveness of advanced point convolutions [122, 193, 193] is also worth to be further investigated.

4.6 Summary

In this chapter, we demonstrated that it is possible to efficiently and effectively segment large-scale point clouds by using a lightweight network architecture. In contrast to most current approaches, that rely on expensive sampling strategies, we instead use random sampling in our framework to significantly reduce the memory footprint and computational cost. A local feature aggregation module is also introduced to effectively preserve useful features from a wide neighbourhood. Extensive experiments on multiple benchmarks demonstrate the high efficiency and the state-of-the-art performance of our approach. It would be interesting to extend our framework for the end-to-end 3D instance segmentation on large-scale point clouds by drawing on the recent work [238] and also for the real-time dynamic point cloud processing [130].

Chapter 5

Learning to Segment Large-Scale 3D Point Clouds with Fewer Labels

5.1 Introduction

In Chapter 4, we presented an efficient neural architecture for semantic segmentation of large-scale 3D point clouds. However, a major challenge in training this algorithm is the requirement for costly point-wise semantic annotation, which can be infeasible and prohibitively time-consuming in real-world scenarios. For example, annotating the entire SensatUrban dataset, introduced in Chapter 3, requires over 600 hours of manual labeling efforts. In this chapter, we aim to address this challenge by proposing a label-efficient solution that reduces the annotation cost while preserving the accuracy of the semantic segmentation. The proposed solution integrates with the previous two chapters to form a comprehensive work flow for large-scale 3D point cloud semantic understanding, encompassing data collection and construction, data labeling, model training, and deployment. The ultimate goal of this thesis is to provide individual users with a complete, feasible solution and work flow for large-scale 3D point cloud semantic understanding.

A comprehensive review of relevant literature reveals that most existing neural architectures for semantic segmentation of 3D point clouds [157, 122, 51, 90, 193, 134, 263, 49] rely on densely annotated per-point semantic labels for training. This reliance, however, corresponds to a significant annotation cost. For instance, annotating the LiDAR-based SemanticKITTI dataset [28] requires over 1700 person-hours, and annotating a single indoor scene ($5\text{m} \times 5\text{m} \times 2\text{m}$) [55] takes approximately 22.3 minutes.

Only a few recent works have attempted to tackle 3D semantic segmentation using limited point labels to train neural networks. These methods can be broadly

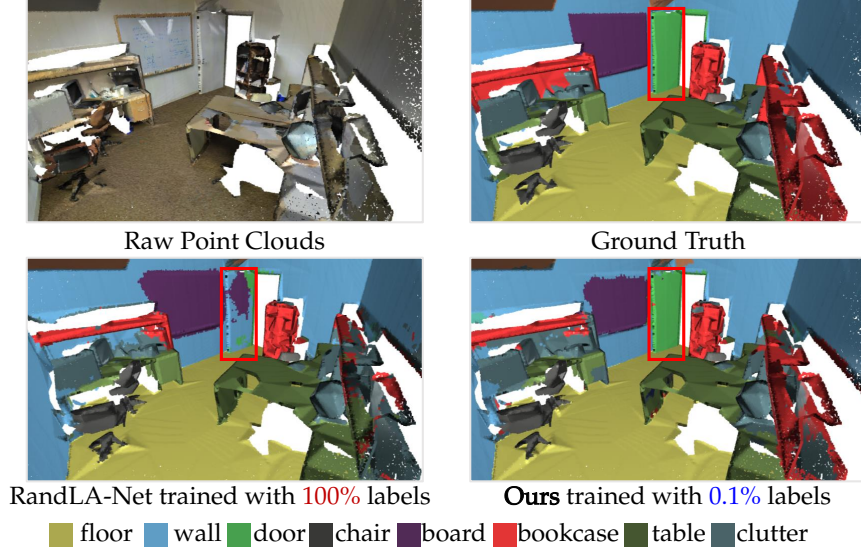


Figure 5.1: Qualitative results of RandLA-Net (Chapter 4) and our SQN on the S3DIS dataset. Trained with only 0.1% annotations, SQN achieves comparable or even better results than the fully-supervised RandLA-Net. Red bounding boxes highlight the superior segmentation accuracy of our SQN.

classified into five categories: 1) Utilizing 2D image labels for training, as demonstrated in [206, 262]; 2) Employing fewer 3D labels through techniques such as gradient approximation, supervision propagation, and perturbation consistency, as seen in [234, 249, 209, 216]; 3) Generating pseudo 3D labels from indirect annotations, as presented in [187, 215]; 4) Using superpoint annotations from over-segmentation, as explored in [187, 48, 133]; and 5) Conducting contrastive pre-training followed by fine-tuning with limited 3D labels, as proposed in [84, 227, 252]. Although these methods have demonstrated encouraging results on multiple datasets, there are still several limitations that need to be addressed.

Firstly, existing approaches usually use custom methods to annotate different amounts of data (*e.g.*, 10%/5%/1% of raw points or superpoints) for training. It is thus unclear what proportion of raw points should be annotated and how, making fair comparison impossible. **Secondly**, to fully utilize the sparse annotations, existing weak-labeling pipelines usually involve multiple stages including careful data augmentation, self-pretraining, fine-tuning, and/or post-processing such as the use of dense CRF [104]. As a consequence, it tends to be more difficult to tune the parameters and deploy them in practical applications, compared with the standard end-to-end training scheme. **Thirdly**, these techniques do not adequately consider the strong local semantic homogeneity of point neighbors in large-scale

point clouds, or do so ineffectively, resulting in the limited, yet valuable, annotations being under-exploited.

Motivated by these issues, we aim to propose a new paradigm for weakly-supervised semantic segmentation on large-scale point clouds, addressing the above shortcomings. In particular, we first explore weak-supervision schemes purely based on existing fully-supervised methods, and take into account two key questions: 1) how does the performance of existing fully-supervised methods change with different amounts of annotated data for training? 2) At what point does the weakly-supervised regime actually begin, given fewer and fewer labels? Our investigation is designed to shed light on the use of mature architectures for this task, as opposed to merely relying on techniques developed for 2D images. Our results indicate that while the accuracy of fully-supervised baselines decreases only slightly when faced with a small amount of random labeled points, beyond a certain threshold, such as 0.1% of the full annotations, the performance degrades significantly.

With this insight, we propose a novel yet simple **Semantic Query Network**, named **SQN**, for semantic segmentation given as few as 0.1% labeled points for training. Our SQN firstly encodes the entire raw point cloud into a set of hierarchical latent representations via an existing feature extractor, and then takes an arbitrary 3D point position as input to query a subset of latent representations within a local neighborhood. These queried representations are summarized into a compact vector and then fed into a series of multilayer perceptrons (MLPs) to predict the final semantic label. Fundamentally, our SQN explicitly and effectively considers the semantic similarity between neighboring 3D points, allowing the extremely sparse training signals to be back-propagated to a much wider spatial region, thereby achieving superior performance under weak supervision.

Overall, this chapter takes a step to bridge the gap between the highly successful fully-supervised methods to the emerging weakly-supervised schemes, in an attempt to reduce the time and labor cost of point-cloud annotation. However, unlike the existing weak-supervision methods, our SQN does not require any self-supervised pretraining, hand-crafted constraints, or complicated post-processing steps, whilst obtaining close to fully-supervised accuracy using as few as 0.1% training labels on multiple large-scale open datasets. Remarkably, for similar accuracy, we find that labeling costs can be reduced by 98% according to our quantitative evaluation in experiments. Figure 5.1 shows the qualitative results of our method. Our key contributions are:

- The observation that existing fully-supervised methods degrade slowly until 1% sparse point annotation, demonstrating that full, dense labeling is not necessary.
- The introduction of a novel weakly supervised method that effectively utilizes sparse training signals through point neighborhood query.
- The demonstration of substantial improvement over baselines in multiple benchmarks, surpassing the current state-of-the-art weak-supervision methods by large margins.

5.2 Exploring Weak Supervision

Since weakly-supervised 3D semantic segmentation is still in its infancy, there is no clear consensus about what are the sensible formulations of weak training signals, and what approach should be used to sparsely annotate a dataset such that a direct comparison is possible. Here, we undertake an initial investigation by conducting a pilot study, examining the performance of existing fully supervised techniques under various formulations of weak training signals.

Weak Annotation Strategy: The fundamental objective of weakly-supervised segmentation is to obtain accurate estimations with as low as possible annotation cost, in terms of labeler time. However, it is non-trivial to compare the cost of different annotation methods in practice. Existing annotation options include: 1) randomly annotating sparse point labels [234, 249, 248], 2) actively annotating sparse point labels [84, 172] or region-wise labels [221], 3) annotating seg-level labels or superpoint labels [187, 133, 48] and 4) annotating sub-cloud labels [215]. All methods have merits. For the purpose of fair reproducibility, we opt for the random point annotation strategy, considering the practical simplicity of building such an annotation tool.

Annotation Tool: To verify the feasibility of random sparse annotations in practice, we develop a user-friendly labeling pipeline based on the off-the-shelf CloudCompare¹ software. Specifically, we first import raw 3D point clouds to the software and randomly downsample them to 10%/1%/0.1% of the total points for sparse annotation. Considering the sparsity of the remaining points, we then explicitly enlarge the size of these sparse points and take the original full point clouds as a reference. As illustrated in Figure 5.2, we then use the standard labeling mode such as polygonal edition for point-wise annotating. (Details and video recordings of our full annotation pipeline are supplied in the experimental section).

¹<https://www.cloudcompare.org/>

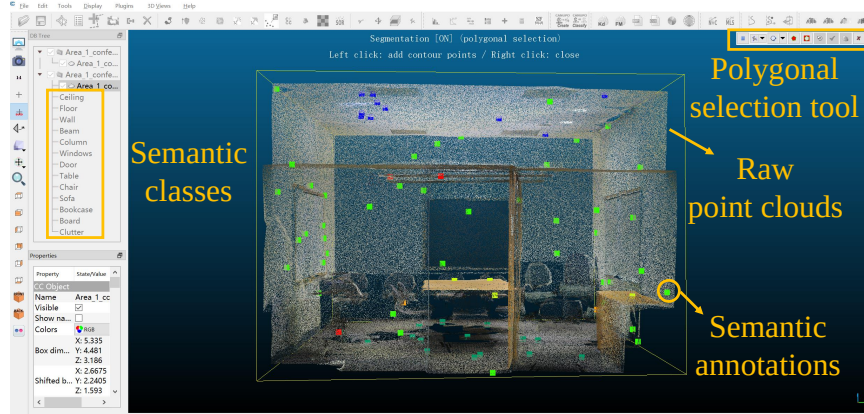


Figure 5.2: Illustration of the sparse annotation tool.

Annotation Cost: With the developed annotation tool, it takes less than 2 minutes to annotate 0.1% of points of a standard room in the S3DIS dataset. For comparison, it requires more than 20 minutes to annotate all points for the same room. Note that, the sparse annotation scheme is particularly suitable for large-scale 3D point clouds with billions of points. As detailed in the experimental section, it only takes about 18 hours to annotate 0.1% of the urban-scale SensatUrban dataset [88], while annotating all points requires more than 600 person-hours [88].

Experimental Settings: We choose the well-known S3DIS dataset [26] as the testbed. The Areas $\{1/2/3/4/6\}$ are selected as the training point clouds, the Area 5 is fully annotated for testing only. With the random sparse annotation strategy, we set up the following four groups of weak signals for training. Specifically, we only annotate the randomly selected 10%/1%/0.1%/0.01% of the 3D points in each room in all training areas.

Using Fully-supervised Methods as Baselines. We select the seminal works PointNet/PointNet++ [155, 157] and the large-scale-point-cloud friendly RandLA-Net introduced in Chapter 4 as baselines. These methods are end-to-end trained on the four groups of weakly annotated data without using any additional modules. Specifically, we follow [90] to conduct the data preparation, and then randomly preserve the semantic labels of a small fraction of points, further training the network by only calculating the loss on these labeled points. The sparse supervision signals are further back-propagated to update the network weights. Ultimately, a total of 12 models (3 models/group $\times$ 4 groups) are trained for evaluation on the full Area 5 of the S3DIS dataset.

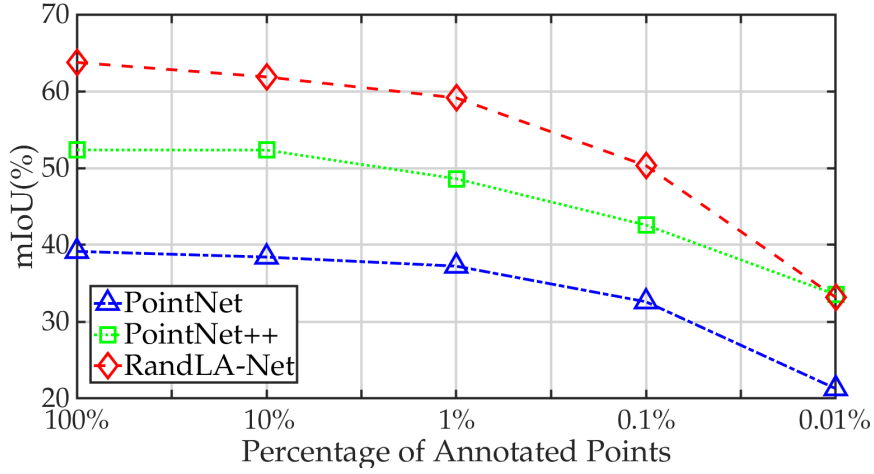


Figure 5.3: Degradation of three baselines in the *Area-5* of S3DIS [26] dataset when decreasing proportions of points that are randomly annotated. (Logarithmic scale used in the horizontal axis).

Results and Findings. Figure 5.3 shows the mIoU scores of all models for segmenting the total 13 classes. The results under full supervision (100% annotations for all training data) are included for comparison. It can be seen that:

- The performance of all baselines only decreases marginally (less than 4%) even though the proportion of point annotations drops significantly from 100% to 1%. This clearly shows that the dense annotations are actually unnecessary to obtain a comparable and favorable segmentation accuracy under the simple random annotation strategy.
- The performance of all baselines drops significantly once the annotated points are less than 0.1%. This critical point indicates that keeping a certain amount of training signals is also essential for weak supervision.

In conclusion, our results demonstrate that utilizing 0.1% of annotated points for training represents a more reasonable weakly supervised setting as compared to utilizing 1% or 5% of annotated points [249, 48]. This leads to the question of how to optimally leverage the sparse but valuable training signals provided by these limited annotations to update the network parameters effectively.

5.3 Semantic Query Network

5.3.1 Overview

Given point clouds with sparse annotations, the fundamental challenge for weakly-supervised learning is how to fully utilize the sparse yet valuable training signals to update the network parameters, such that more geometrically meaningful local patterns can be learned. To resolve this, we design a simple SQN which consists of two major components: 1) a point local feature extractor to learn diverse visual patterns; 2) a flexible point feature query network to collect as many as possible relevant semantic features for weakly-supervised training. As shown in Figure 5.4, our two sub-networks are illustrated by the stacked blocks.

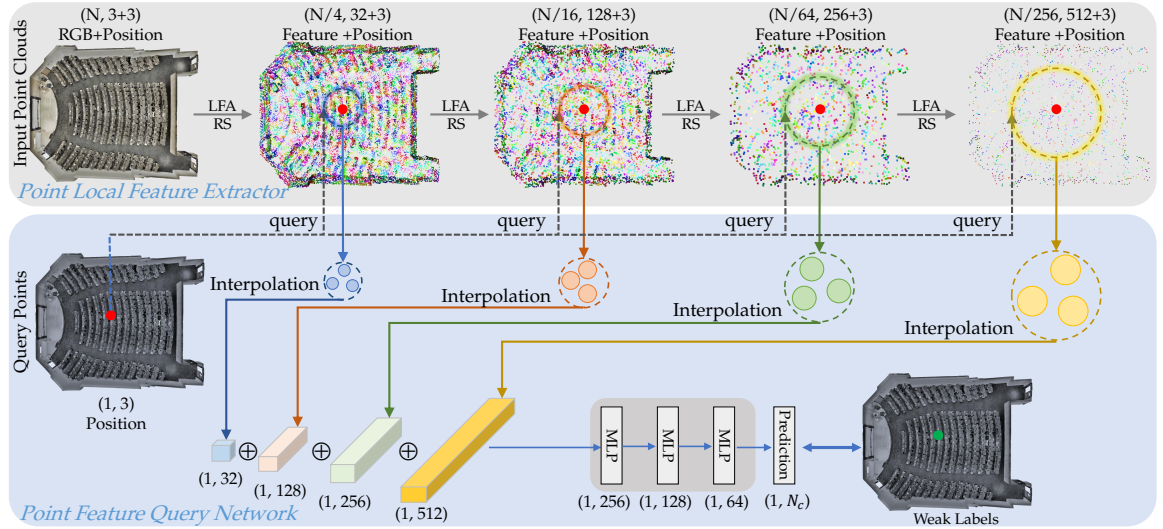


Figure 5.4: The pipeline of our SQN at the training stage with weak supervision. We only show one query point for simplicity.

5.3.2 Point Local Feature Extractor

This component aims to extract local features for all points. As discussed in Section 2.2, there are many excellent backbone networks that are able to extract per-point features. In general, these networks stack multiple encoding layers together with downsampling operations to extract hierarchical local features. In this chapter, we use the encoder of RandLA-Net (Chapter 4) as our feature extractor thanks to its efficiency on large-scale point clouds, although our SQN is not restricted to any particular backbone network.

As shown in the top block of Figure 5.4, the encoder includes four layers of Local Feature Aggregation (LFA) followed by a Random Sampling (RS) operation. Details refer to RandLA-Net (Chapter 4). Given an input point cloud $\mathcal{P}$ with N points, four levels of hierarchical point features are extracted after each encoding layer, *i.e.*, 1) $\frac{N}{4} \times 32$, 2) $\frac{N}{16} \times 128$, 3) $\frac{N}{64} \times 256$, and 4) $\frac{N}{256} \times 512$. To facilitate the subsequent query network, the corresponding point location xyz are always preserved for each hierarchical feature vector.

5.3.3 Point Feature Query Network

Given the extracted point feature vectors, this query network is designed to collect as many relevant features, to be trained using the available sparse signals. In particular, as shown in the bottom block of Figure 5.4, it takes a specific 3D query point as input and then acquires a set of learned point features relevant to that point. Fundamentally, this is assumed that the query point shares similar semantic information with the collected point features, such that the training signals from the query points can be shared and back-propagated for the relevant points. The network consists of three modules: 1) Searching Spatial Neighbouring Point Features, 2) Interpolating Query Point Features, 3) Inferring Query Point Semantics.

Searching Spatial Neighbouring Point Features. Given a 3D query point p with its location xyz , this module is to simply search the nearest K points in each of the previous 4-level encoded feature vectors, this can be achieved by utilizing the K nearest neighbor search, based on the point-wise Euclidean distance. For example, as to the first level of extracted point features, the most relevant K points are selected, acquiring the raw feature vectors $\{\mathbf{F}_p^1, \dots, \mathbf{F}_p^K\}$.

Interpolating Query Point Features. For each level of features, the queried K vectors are compressed into a compact representation for the query point p . For simplicity, we apply the trilinear interpolation method to compute a feature vector for p , according to the Euclidean distance between p and each of K points. Eventually, four hierarchical feature vectors are concatenated together, representing all relevant point features from the entire 3D point cloud.

Inferring Query Point Semantics. After obtaining the unique and representative feature vector for the query point p , we feed it into a series of MLPs, directly inferring the point semantic category.

Overall, given a sparse number of annotated points, we parallelly query their neighboring point features for training. This allows the valuable training signals to be back-propagated to a much wider spatial context. During testing, all 3D points

are fed into the two sub-networks for semantic estimation. In fact, our simple query mechanism greatly enables the network to infer the point semantic category from a significantly larger receptive field. To summarize, there are two key differences between the semantic query mechanism of SQN and existing (fully-supervised) architectures, which can be summarized as follows:

- **Wider Semantic Propagation:** Given sparse labels, the key is to effectively spread the limited supervision signals to wider contexts. SQN fully exploits the prior knowledge of local semantic homogeneity, by querying parallelly and hierarchically at wider and wider spatial contexts.
- **Easily Learnable Architecture:** Existing fully-supervised networks usually adopt dense and deep U-Net connections. Despite sophistication, they are data-hungry and can be hardly well optimized, particularly in settings of extremely sparse labels. However, our light SQN simply adopts encoding layers followed by MLPs, allowing the parameters to be easily learned with sparse signals.

5.3.4 Implementation Details

The hyper-parameter K is set to 3 for semantic query in our SQN framework and kept consistent for all experiments. Our SQN is trained end-to-end with 0.1% randomly annotated points. All experiments are conducted on a PC with an Intel Core™ i9-10900X CPU and an NVIDIA RTX Titan GPU with 24G memory. Note that, the proposed SQN framework allows flexible use of different backbone networks such as voxel-based MinkowskiNet [51], please refer to the experiment part for more details.

5.3.5 Relations to Previous SuperPoint-based Methods

Here we discuss the relation between the proposed SQN and the previous superpoint-based methods [48, 133]. To clarify, both SQN and superpoint approaches have similar design philosophies of leveraging local semantic homogeneity, but in different ways. The key difference can be summarized as follows:

- The key idea of superpoint methods is to partition the raw point clouds into geometrically simple primitives (*i.e.*, **hard** superpoints). However, such a strong assumption collapses on sparse LiDAR point clouds. This can be verified by the fact that the mIoU of SPGraph on SemanticKITTI is only 17.4%

(Table 4). Not surprisingly, the superpoint-based weakly supervised semantic segmentation network [48, 133] all fail to report experimental results on SemanticKITTI.

- By contrast, SQN uses a simple and flexible query mechanism followed by vanilla MLPs, thus learning **soft** embedding of *superpoints* or *superareas* from sparse annotations. We also evaluate the segmentation performance of our SQN on the SemanticKITTI dataset (Sec. 5.4.4.4), which is composed of sparse LiDAR point clouds. Hence, SQN is more general and effective for different types of point clouds.

5.4 Experiments

In this section, we first introduce the developed annotation tool for sparse point clouds and the detailed annotation cost of our sparse annotation scheme. Next, we compare the semantic segmentation performance of our SQN with the state-of-the-art weakly supervised methods on the S3DIS [26], ScanNet [55], and Semantic3D [76] benchmarks. Then, we evaluate our SQN on four large-scale point clouds datasets including SensatUrban [88], Toronto3D [185], DALES [200], and SemanticKITTI [28]. Lastly, ablation studies are conducted to evaluate the effectiveness of each component.

5.4.1 Sparse Annotation Tool

5.4.1.1 Annotation Pipeline

As mentioned in Section 5.2, we develop a user-friendly annotation pipeline based on the off-the-shelf software. Here, we provide more detailed information on the pipeline. Specifically, given large-scale raw point clouds, the sparse annotation pipeline could be generally divided into the following steps:

1. Load the raw point clouds;
2. Random downsample to a specified ratio (e.g., 0.1%);
3. Increase the point size of the downsampled points;
4. Visualize the original point cloud and the down-sampled point cloud simultaneously;
5. Annotate downsampled points in polygonal edition mode;
6. Refine point labels.

We also provide a video illustrating the annotation pipeline, which can be viewed at <https://youtu.be/N0UAeY3lmsY>.

5.4.1.2 The Number of Annotated Points on 7 Datasets

Considering that the existing large-scale point cloud datasets usually have millions of points, and typically have relatively high density, we therefore follow [193, 90] to perform grid downsampling of raw point clouds at the beginning, and then execute the random-based annotation steps in practice. Note that, all experiments of our SQN on the seven public datasets follow this setting. As shown in Table 5.1, the grid downsampling at the beginning can significantly reduce the number of raw points. Taking the Semantic3D dataset which has high density as an example, the total number of points after grid downsampling becomes 1/50 of the original point clouds. Following the 0.1% sparse annotation pipeline in our SQN, the total number of annotated points is only 78100, which is approximately 0.002% of the total raw points. To avoid confusion, we still report the 0.1% labeling ratio in the main paper to keep consistency (*i.e.*, the number of annotated points after grid downsampling / the total number of points after grid downsampling). Importantly, this is significantly different from 1T1C [133] and cannot be directly compared, which calculates its labeling ratio by using the number of labeled instances divided by the total number of points, so as to achieve an over-exaggerated labeling ratio.

Table 5.1: A comparison of the total number of points (M: Million) before and after grid sampling for seven public datasets. The grid size and the number of actual annotated points under our 0.1% supervision setting are also reported.

	Grid size	Raw pts	Grid sampled pts	Anno. pts (0.1%)
S3DIS [26]	0.04	273M	18.6M	18,600
Semantic3D [76]	0.06	4000M	78.1M	78,100
ScanNet [200]	0.04	242M	60.2M	60,200
SemanticKITTI [28]	0.06	5299M	3401M	3.4M
DALES [200]	0.32	505M	211M	211,000
SensatUrban [88]	0.2	2847M	221M	221,000
Toronto3D [185]	0.04	78.3M	24.3M	24,300

5.4.1.3 Annotation Cost

The sparse annotation scheme used in our SQN can greatly reduce the annotation cost in practice, especially for extremely large-scale 3D point clouds with billions

of points. Taking 0.1% sparse point annotation as an example, with the developed CloudComapre-based labeling tool, a professional annotator can finish the annotation of the whole SensatUrban [88] dataset within 16 hours. By comparison, the original dense point-wise labeling costs 600 person-hours. Primarily, *this is because the random annotation-based pipeline offers great error tolerance to avoid annotating boundary areas* (as only a very small number of points fall on the boundary), hence advanced functions such as polygonal edition can be freely and flexibly use, finally improve the productivity. In the traditional dense labeling pipeline, annotators are usually required to rotate and zoom back and forth to accurately separate the boundary areas, which consumes most of the labeling time. However, the random annotation-based pipeline used in our developed tool can greatly reduce such time-consuming labeling of boundary areas, hence greatly reducing the overall annotation cost. Note that, the annotation cost (*i.e.*, the total annotation time) could be further reduced if more advanced annotation software such as QTModeler² is used, where the user interface is more friendly.

5.4.2 Experimental Setup

(1) Data augmentation. We follow [234] to apply different data augmentation techniques on the input point clouds during training, including random flipping, random rotation, and random noise.

(2) Re-training with generated pseudo labels. We observe that different datasets (*e.g.*, S3DIS [26] *vs.* Semantic3D [76]) have significantly different number of total points (273 million *vs.* 4000 million points). Therefore, the actual number of annotated points under our weak supervision setting (0.1%) are also different (18600 *vs.* 78100, as reported in Table 5.1). In the experiment, for the relatively small-scale S3DIS dataset which has extremely sparse supervision signals, we empirically find that retraining a new model with the generated pseudo labels can further increase the final segmentation performance. In particular, we firstly train our SQN with the limited annotated 0.1% points, and then infer the semantics of the entire training set. These estimated semantics are regarded as pseudo labels. After that, we retrain a new model of our SQN from scratch with the generated pseudo labels. This retraining trick is able to fully utilize the extremely limited but valuable supervision signals. However, for large-scale datasets including Semantic3D [76], SensatUrban [88], SemanticKITTI [28], DALES [200] in Section 5.4.4, our SQN can

²<https://appliedimagery.com/>

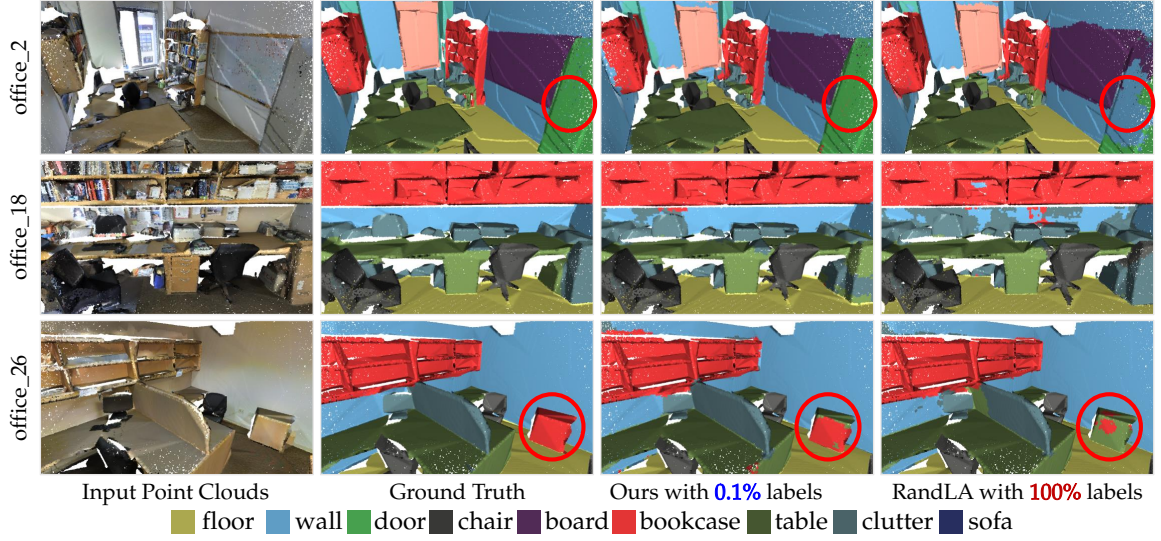


Figure 5.5: Qualitative results achieved by our SQN and the fully-supervised RandLA-Net (Chapter 4) on the *Area-5* of the S3DIS dataset.

achieve satisfactory results trained with 0.1% annotated points, while the retraining trick does not noticeably improve the performance. Advanced techniques such as pseudo label refining [246] will be further explored in future work.

5.4.3 Comparison with SOTA Approaches

We first evaluate the performance of our SQN on three commonly-used benchmarks including S3DIS [26], ScanNet [55] and Semantic3D [76]. Following [90], we use the Overall Accuracy (OA) and mean Intersection-over-Union (mIoU) as the main evaluation metrics. Note that, we evaluate the final performance on all points of the original test set for a fair comparison.

5.4.3.1 Evaluation on S3DIS

The indoor S3DIS [26] dataset has 273 million 3D points belonging to 13 classes. Following [234], we report the results on Area-5 in Table 5.2. Note that, our SQN is compared with three groups of approaches: 1) Fully-supervised methods including SPGraph [110], KPConv [193] and RandLA-Net with 100% training labels; 2) Weakly supervised approaches that learn from limited superpoint annotations including 1T1C [133] and SSPC-Net [48]; 3) Weakly-supervised methods [234, 188, 107] that learning from limited annotations. We also list the proportion of annotations used for training.

Table 5.2: Quantitative results of different methods on the *Area-5* of S3DIS dataset. Mean IoU (mIoU, %), and per-class IoU (%) scores are reported. Note that, [†]The ratios of these methods represent the ratio of super-point annotations and cannot be directly compared. Bold represents the best result in weak supervision and underlined represents the best in full supervision.

	Methods	mIoU(%)	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
Full supervision	PointNet [155]	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	58.9	52.6	5.9	40.3	26.4	33.2
	PointCNN [122]	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
	SPGraph [110]	58.0	89.4	96.9	78.1	0.0	2.8	48.9	61.6	4.7	75.4	69.8	52.6	2.1	52.2
	SPH3D [116]	59.5	3.3	97.1	81.1	0.0	33.2	45.8	43.8	79.7	86.9	33.2	71.5	54.1	53.7
	PointWeb [255]	60.3	92.0	8.5	79.4	0.0	21.1	59.7	34.8	76.3	88.3	46.9	69.3	64.9	52.5
	RandLA-Net [90]	63.0	92.4	96.7	80.6	0.0	18.3	1.3	43.3	77.2	85.2	1.5	71.0	9.2	52.3
	KPCConv rigid [193]	5.4	92.6	97.3	1.4	0.0	16.5	54.5	9.5	80.2	0.1	66.4	4.6	63.7	8.1
Limited superpoint labels [†]	1T1C (0.02%) [133]	50.1	-	-	-	-	-	-	-	-	-	-	-	-	-
	SSPC-Net (0.01%) [48]	51.5	-	-	-	-	-	-	-	-	-	-	-	-	-
Limited point-wise labels	II Model (10%) [107]	46.3	91.8	97.1	73.8	0.0	5.1	42.0	19.6	67.2	66.7	47.9	19.1	30.6	41.3
	MT (10%) [188]	47.9	92.2	96.8	74.1	0.0	10.4	46.2	17.7	70.7	67.0	50.2	24.4	30.7	42.2
	Xu (10%) [234]	48.0	90.9	97.3	74.8	0.0	8.4	49.3	27.3	71.7	69.0	53.2	16.5	23.3	42.8
	Zhang et al. (1%) [248]	61.8	91.5	96.9	80.6	0.0	18.2	58.1	47.2	75.8	85.7	65.2	68.9	65.0	50.2
	PSD (1%) [249]	63.5	92.3	97.7	80.7	0.0	27.8	56.2	62.5	78.7	84.1	63.1	70.4	58.9	53.2
	II Model (0.2%) [107]	44.3	89.1	97.0	71.5	0.0	3.6	43.2	27.4	63.1	62.1	43.7	14.7	24.0	36.7
	MT (0.2%) [188]	44.4	88.9	96.8	70.1	0.1	3.0	44.3	28.8	63.7	63.6	47.7	15.5	23.0	35.8
	Xu (0.2%) [234]	44.5	90.1	97.1	71.9	0.0	1.9	47.2	29.3	64.0	62.9	42.2	15.9	18.9	37.5
	RandLA-Net (0.1%)	52.9	89.9	95.9	75.3	0.0	7.5	52.4	26.5	62.2	74.5	49.1	60.2	49.3	45.1
	Ours (0.1%)	61.4	91.7	95.6	78.7	0.0	24.2	55.8	63.1	70.5	83.1	60.6	67.8	56.1	50.6

Table 5.3: Quantitative results of different approaches on ScanNet (*online test set*). Mean IoU (mIoU, %), and per-class IoU (%) scores are reported.

Settings	Method	mIoU(%)	bath	bed	bkshf	cab	chair	cntr	curt	desk	door	floor	other	pic	fridge	show	sink	sofa	table	toil	wall	wind
Full supervision	ScanNet [55]	30.6	20.3	36.6	50.1	31.1	52.4	21.1	0.2	34.2	18.9	78.6	14.5	10.2	24.5	15.2	31.8	34.8	30.0	46.0	43.7	18.2
	PointNet++ [157]	33.9	58.4	47.8	45.8	25.6	36.0	25.0	24.7	27.8	26.1	67.7	18.3	11.7	21.2	14.5	36.4	34.6	23.2	54.8	52.3	25.2
	SPLATNet3D [179]	39.3	47.2	51.1	60.6	31.1	65.6	24.5	40.5	32.8	19.7	92.7	22.7	0.0	0.1	24.9	27.1	51.0	38.3	59.3	69.9	26.7
	Tangent-Conv [189]	43.8	43.7	64.6	47.4	36.9	64.5	35.3	25.8	28.2	27.9	91.8	29.8	14.7	28.3	29.4	48.7	56.2	42.7	61.9	63.3	35.2
	PointCNN [122]	45.8	57.7	61.1	35.6	32.1	71.5	29.9	37.6	32.8	31.9	94.4	28.5	16.4	21.6	22.9	48.4	54.5	45.6	75.5	70.9	47.5
	PointConv [222]	55.6	63.6	64.0	57.4	47.2	73.9	43.0	43.3	41.8	44.5	94.4	37.2	18.5	46.4	57.5	54.0	63.9	50.5	82.7	76.2	51.5
	SPH3D-GCN [116]	61.0	85.8	77.2	48.9	53.2	79.2	40.4	64.3	57.0	50.7	93.5	41.4	4.6	51.0	70.2	60.2	70.5	54.9	85.9	77.3	53.4
	KPCConv [193]	68.4	84.7	75.8	78.4	64.7	81.4	47.3	77.2	60.5	59.4	93.5	45.0	18.1	58.7	80.5	69.0	78.5	61.4	88.2	81.9	63.2
	SparseConvNet [73]	72.5	64.7	82.1	84.6	72.1	86.9	53.3	75.4	60.3	61.4	95.5	57.2	32.5	71.0	87.0	72.4	82.3	62.8	93.4	86.5	68.3
	SegGCN [115]	58.9	83.3	73.1	53.9	51.4	78.9	44.8	46.7	57.3	48.4	93.6	39.6	6.1	50.1	50.7	59.4	70.0	56.3	87.4	77.1	49.3
	RandLA-Net [90]	64.5	77.8	73.1	69.9	57.7	82.9	44.6	73.6	47.7	52.3	94.5	45.4	26.9	48.4	74.9	61.8	73.8	59.9	82.7	79.2	62.1
	Ours (0.1%)	56.9	67.6	69.6	65.7	49.7	77.9	42.4	54.8	51.5	37.6	90.2	42.2	35.7	37.9	45.6	59.6	65.9	54.4	68.5	66.5	55.6

It can be seen that our SQN outperforms other weakly-supervised methods including [188, 234, 107] by large margins with only 0.1% annotations used, and surpasses the baseline RandLA-Net by nearly 9% under same weakly-supervised settings. In particular, our SQN is also comparable to the fully-supervised RandLA-Net. Figure 5.5 shows qualitative comparisons of RandLA-Net and our SQN.

5.4.3.2 Evaluation on ScanNet

The ScanNet [55] dataset consists of 1613 indoor scans (1201 for training, 312 for validation, and 100 for online testing). It has nearly 242 million points sampled from the densely reconstructed 3D meshes. We report the quantitative results achieved by different approaches on the hidden test set in Table 5.3. It can be seen that our SQN achieves higher mIoU scores with only 0.1% training labels, compared with MPRM [215] which is trained with sub-cloud labels, and Zhang et al. [248] and PSD [249] trained with 1% annotations.

Table 5.4: Quantitative results of different approaches on Semantic3D (*semantic-8*) [76]. This test consists of 2,091,952,018 points. The scores are obtained from recent publications. Bold represents the best result in weakly-supervised methods, and underlined represents the best results in fully-supervised methods.

Settings	Methods	mIoU(%)	OA(%)	<i>man-made.</i>	<i>natural.</i>	<i>high veg.</i>	<i>low veg.</i>	<i>buildings</i>	<i>hard scape</i>	<i>scanning art.</i>	<i>cars</i>
Full supervision	TML-PC [147]	39.1	74.5	80.4	66.1	42.3	41.2	64.7	12.4	0.0	5.8
	TMLC-MS [77]	49.4	85.0	91.1	69.5	32.8	21.6	87.6	25.9	11.3	55.3
	PointNet++ [157]	63.1	85.7	81.9	78.1	64.3	51.7	75.9	36.4	43.7	72.6
	EdgeConv [53]	64.4	89.6	91.1	69.5	65.0	56.0	89.7	30.0	43.8	69.7
	SnapNet [32]	67.4	91.0	89.6	79.5	74.8	56.1	90.9	36.5	34.3	77.2
	PointGCR [139]	69.5	92.1	93.8	80.0	64.4	66.4	93.2	39.2	34.3	85.3
	RGNet [195]	72.0	90.6	86.4	70.3	69.5	68.0	96.9	43.4	52.3	89.5
	LCP [33]	74.6	94.1	94.7	85.2	77.4	70.4	94.0	52.9	29.4	92.6
	SPGraph [110]	76.2	92.9	91.5	75.6	78.3	71.7	94.4	56.8	52.9	88.4
	ConvPoint [31]	76.5	93.4	92.1	80.6	76.0	71.9	95.6	47.3	61.1	87.7
	RandLA-Net [90]	75.8	95.0	<u>97.4</u>	<u>93.0</u>	70.2	<u>65.2</u>	94.4	49.0	<u>44.7</u>	92.7
	WreathProdNet [211]	77.1	94.6	95.2	87.1	75.3	67.1	96.1	51.3	51.0	93.4
Weak supervision	Ours (0.1%)	72.3	94.8	97.9	93.2	65.5	63.4	94.9	44.9	47.4	70.9
	Ours (0.01%)	58.8	91.9	96.7	90.3	56.6	53.3	90.7	13.6	24.0	44.9

Table 5.5: Quantitative results of different approaches on Semantic3D (*reduced-8*) [76]. This test consists of 78,699,329 points. The scores are obtained from the recent publications. Bold represents the best result in weakly-supervised methods, and underlined represents the best results in fully-supervised methods.

Settings	Methods	mIoU(%)	OA(%)	<i>man-made.</i>	<i>natural.</i>	<i>high veg.</i>	<i>low veg.</i>	<i>buildings</i>	<i>hard scape</i>	<i>scanning art.</i>	<i>cars</i>
Full supervision	SnapNet [32]	59.1	88.6	82.0	77.3	79.7	22.9	91.1	18.4	37.3	64.4
	SEGCloud [190]	61.3	88.1	83.9	66.0	86.0	40.5	91.1	30.9	27.5	64.3
	RF_MSSF [192]	62.7	90.3	87.6	80.3	81.8	36.4	92.2	24.1	42.6	56.6
	MSDeepVoxNet [166]	65.3	88.4	83.0	67.2	83.8	36.7	92.4	31.3	50.0	78.2
	ShellNet [254]	69.3	93.2	96.3	90.4	83.9	41.0	94.2	34.7	43.9	70.2
	GACNet [208]	70.8	91.9	86.4	77.7	<u>88.5</u>	<u>60.6</u>	94.2	37.3	43.5	77.8
	SPG [110]	73.2	94.0	97.4	92.6	87.9	44.0	83.2	31.0	63.5	76.2
	KPConv [193]	74.6	92.9	90.9	82.2	84.2	47.9	94.9	40.0	<u>77.3</u>	<u>79.9</u>
	RGNet [195]	74.7	94.5	<u>97.5</u>	<u>93.0</u>	88.1	48.1	94.6	36.2	<u>72.0</u>	<u>68.0</u>
	RandLA-Net [90]	77.4	94.8	95.6	91.4	86.6	51.5	95.7	51.5	69.8	76.8
Weak supervision	Ours (0.1%)	74.7	93.7	97.1	90.8	84.7	48.5	93.9	37.4	71.0	74.5
	Ours (0.01%)	65.6	90.3	96.6	87.5	80.6	37.1	88.5	16.9	56.6	60.9

5.4.3.3 Evaluation on Semantic3D

This dataset consists of 30 urban and rural street scenarios (15 for training and 15 for online testing). There are 4 billion points in total acquired by the terrestrial laser. The detailed experimental results achieved on the *Semantic8* and *Reduced8* subset of the Semantic3D dataset are reported in Table 5.4 and Table 5.5. It can be seen that our SQN trained with 0.1% labels achieves competitive performance with fully-supervised baselines on both *Semantic8* and *Reduced8* subsets. This clearly demonstrates the effectiveness of our semantic query framework, which takes full advantage of the limited annotations. Additionally, we also train our SQN with only 0.01% randomly annotated points, considering the extremely large amount of 3D points scanned. We can see that our SQN trained with 0.01% labels also achieves satisfactory accuracy, though there is space to be improved in the future. In addition, we also show the qualitative results achieved by our SQN on

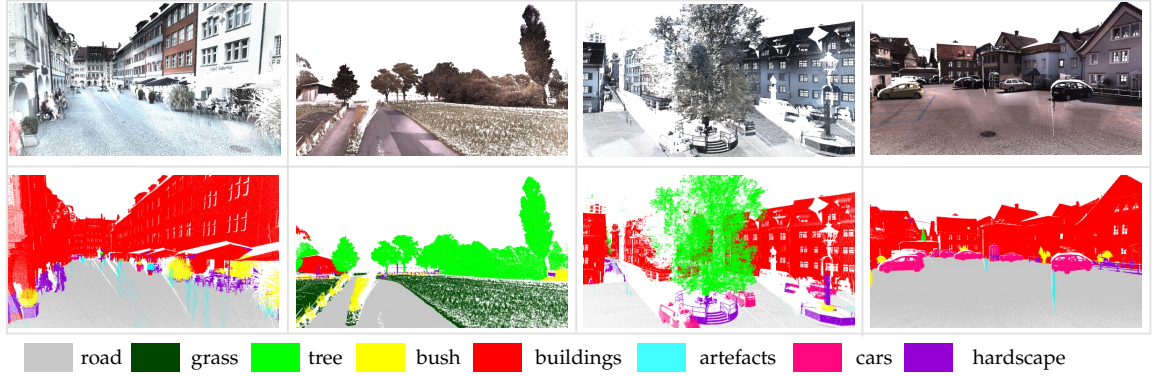


Figure 5.6: Qualitative results achieved by our SQN on the reduced-8 split of Semantic3D. Note that, the ground truth of the test set is not publicly available.

the *Reduced-8* subset with 0.1% labels in Fig 5.6.

5.4.4 Evaluation on Large-Scale 3D Benchmarks

To validate the versatility of our SQN, we further evaluate our SQN on four outdoor large-scale point cloud datasets with significant different density and quality, including SensatUrban [88], Toronto3D [185], DALES [200], and SemanticKITTI [28]. Note that, all existing weakly supervised approaches are only evaluated on the dataset with dense point clouds, and there are no results reported on these datasets. Therefore, we only compare our approach with existing fully-supervised methods in this section.

5.4.4.1 Evaluation on SensatUrban

This is a new urban-scale photogrammetry point cloud dataset covering over 7.6 square kilometers of urban areas in the UK. It has nearly 3 billion points in total. Note that, this dataset is extremely challenging due to the imbalanced class distributions. The detailed experimental results achieved on the SensatUrban dataset are reported in Table 5.6. In addition, we also show the qualitative results achieved by our SQN trained with only 0.1% labels on this dataset in Fig. 5.11.

5.4.4.2 Evaluation on Toronto3D

This dataset consists of 1KM urban road point clouds acquired by vehicle-mounted mobile laser systems. It has 78.3 million points belonging to 8 semantic categories. Here, we provide the quantitative comparison of our SQN and several

Table 5.6: Benchmark results of the baselines on our SensatUrban. Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%) scores are reported. Bold represents the best result in weakly-supervised methods, and underlined represents the best results in fully-supervised methods.

Settings	Methods	OA(%)	mAcc(%)	mIoU(%)	ground	veg.	building	wall	bridge	parking	rail	traffic	street.	car	footpath	bike	water
Full supervision	PointNet [155]	80.78	30.32	23.71	67.96	89.52	80.05	0.00	0.00	3.95	0.00	31.55	0.00	35.14	0.00	0.00	0.00
	PointNet++ [157]	84.30	39.97	32.92	72.46	94.24	84.77	2.72	2.09	25.79	0.00	31.54	11.42	38.84	7.12	0.00	56.93
	TagentConv [189]	76.97	43.71	33.30	71.54	91.38	75.90	35.22	0.00	45.34	0.00	26.69	19.24	67.58	0.01	0.00	0.00
	SPGraph [110]	85.27	44.39	37.29	69.93	94.55	88.87	32.83	12.58	15.77	<u>15.48</u>	30.63	22.96	56.42	0.54	0.00	44.24
	SparseConv [73]	88.66	63.28	42.66	74.10	97.90	94.20	63.30	7.50	24.20	0.00	30.10	34.00	74.40	0.00	0.00	54.80
	KPConv [193]	93.20	63.76	57.58	87.10	98.91	95.33	74.40	28.69	41.38	0.00	55.99	54.43	85.67	40.39	0.00	86.30
	RandLA-Net [90]	89.78	69.64	52.69	80.11	98.07	91.58	48.88	40.75	51.62	0.00	56.67	33.23	80.14	32.63	0.00	71.31
Weak supervision	Ours (0.1%)	90.97	70.84	53.97	83.41	98.22	94.22	48.38	50.84	40.89	14.53	50.72	38.48	75.62	34.03	0.00	72.26
	Ours (0.01%)	85.57	49.40	37.17	74.89	96.67	88.77	32.43	7.49	12.84	0.00	29.32	22.15	67.25	0.02	0.00	51.38

Table 5.7: Quantitative results of different approaches on the Toronto3D [185] dataset. The scores of the baselines are obtained from [185]. Bold represents the best result in weakly-supervised methods, and underlined represents the best results in fully-supervised methods.

Settings	Methods	OA(%)	mIoU(%)	Road	Rd mrk.	Natural	Building	Util. line	Pole	Car	Fence
Full supervision	PointNet++ [157]	84.88	41.81	89.27	0.00	69.06	54.16	43.78	23.30	52.00	2.95
	PointNet++ (MSG) [157]	92.56	59.47	92.90	0.00	86.13	82.15	60.96	62.81	76.41	14.43
	DGCNN [213]	94.24	61.79	93.88	0.00	91.25	80.39	62.40	62.32	88.26	15.81
	KPFCNN [193]	95.39	69.11	94.62	0.06	96.07	91.51	87.68	<u>81.56</u>	85.66	15.72
	MS-PCNN [138]	90.03	65.89	93.84	3.83	93.46	82.59	67.80	71.95	91.12	22.50
	TGNet [123]	94.08	61.34	93.54	0.00	90.83	81.57	65.26	62.98	88.73	7.85
	MS-TGNet [185]	95.71	70.50	94.41	17.19	95.72	88.83	76.01	73.97	<u>94.24</u>	23.64
	RandLA-Net (w/ RGB) [†] [90]	97.15	81.88	<u>96.69</u>	<u>64.10</u>	96.85	<u>94.14</u>	88.03	77.48	<u>93.21</u>	44.53
	RandLA-Net (w/o RGB) [90]	<u>95.63</u>	<u>77.72</u>	<u>94.53</u>	<u>42.44</u>	96.62	<u>93.10</u>	<u>86.56</u>	76.83	92.55	39.14
Weak supervision	Ours (w/ RGB, 0.1%)[†]	96.67	77.75	96.69	65.67	94.58	91.34	83.36	70.59	88.87	30.91
	Ours (w/o RGB, 0.1%)	92.84	69.35	93.74	16.83	92.55	89.04	82.50	63.98	88.17	28.01
	Ours (w/ RGB, 0.01%)[†]	94.19	68.17	95.26	54.44	88.20	84.07	75.87	57.52	84.33	5.69
	Ours (w/o RGB, 0.01%)	90.47	57.57	90.97	4.99	84.10	80.29	62.78	56.51	69.49	11.44

fully-supervised methods in Table 5.7. Following [90], we also additionally report the performance of our method with and without color information. It can be seen that our SQN outperforms several fully-supervised methods such as KPConv, with merely 0.1% of point annotations for training. We also notice that the usage of color information closes the gap between our method and the top-performing RandLA-Net. This implies that it could be helpful to introduce auxiliary information under the setting of weak supervision.

5.4.4.3 Evaluation on DALES

This dataset consists of large-scale earth scans acquired by an aerial LiDAR. It covers over 10 km^2 spatial ranges with 5 million points belonging to 8 semantic categories. We compare our SQN with strong fully-supervised approaches. As shown in Table 5.8, our method achieves higher mIoU scores than PointNet++ [157], ConvPoint [33], SPGraph [110], PointCNN [122] and ShellNet [254], with only 0.1%

Table 5.8: Quantitative results of different approaches on the DALES dataset. Overall Accuracy (OA, %), mean class Accuracy (mAcc, %), mean IoU (mIoU, %), and per-class IoU (%) are reported. Bold represents the best result in weakly-supervised methods, and underlined represents the best results in fully-supervised methods.

Settings	Method	OA(%)	mIoU(%)	<i>ground</i>	<i>buildings</i>	<i>cars</i>	<i>trucks</i>	<i>poles</i>	<i>power lines</i>	<i>fences</i>	<i>vegetation</i>
Full supervision	ShellNet [254]	96.4	57.4	96.0	95.4	32.2	39.6	20.0	27.4	60.0	88.4
	PointCNN [122]	97.2	58.4	97.5	95.7	40.6	4.80	57.6	26.7	52.6	91.7
	SuperPoint [110]	95.5	60.6	94.7	93.4	62.9	18.7	28.5	65.2	33.6	87.9
	ConvPoint [31]	97.2	67.4	96.9	96.3	75.5	21.7	40.3	86.7	29.6	91.9
	PointNet++ [157]	95.7	68.3	94.1	89.1	75.4	30.3	40.0	79.9	46.2	91.2
	KPCConv [193]	97.8	81.1	97.1	96.6	85.3	41.9	75.0	95.5	63.5	94.1
	RandLA-Net [90]	97.1	80.0	97.0	93.2	83.7	43.8	59.4	94.8	71.5	96.6
	Pyramid Point [201]	98.3	83.6	97.8	97.3	88.4	47.9	77.6	96.7	67.5	95.4
Weak supervision	Ours (0.1%)	97.1	72.0	96.7	92.0	75.2	27.3	87.4	48.1	53.7	95.8
	Ours (0.01%)	95.9	60.4	95.9	90.1	57.7	12.8	75.2	32.9	24.9	93.4

labels for training. However, there is still a performance gap compared with the leading fully-supervised counterparts such as RandLA-Net, primarily due to our weak performance on minor categories such as *trucks* and *cars*. The potential reason is that the simple random annotation strategy may happen to ignore the underrepresented classes.

5.4.4.4 Evaluation on SemanticKITTI

This large-scale dataset consists of point cloud sequences captured by LiDAR for autonomous driving. In particular, it has 22 sequences, 43552 sparse scans, and nearly 4 billion points. Note that, RGB is not available in this dataset. We compare our SQN with fully-supervised techniques on the online test set in Table 5.9. It can be seen that our approach achieves a satisfactory mIoU score, outperforming several strong baselines with only 0.1% labels for training. In addition, our model only has 1.05 million trainable parameters, and is extremely lightweight and suitable for real-world applications. Finally, we also visualize the segmentation results achieved by our SQN on the validation set of the SemanticKITTI dataset in Fig. 5.7.

5.4.5 Ablation Study

To evaluate the effectiveness of each module in our framework, we conduct the following ablation studies. All ablated networks are trained on Areas{1/2/3/4/6} with 0.1% labels, and tested on the *Area-5* of the S3DIS dataset.

Table 5.9: Quantitative results of different approaches on SemanticKITTI [28]. The scores are obtained from recent publications. Bold represents the best result in weakly-supervised methods, and underlined represents the best results in fully-supervised methods.

			mIoU(%)	Params(M)	road	sidewalk	parking	other-ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic-sign
Settings	Methods																						
Full supervision	PointNet [155]	14.6	3	61.6	35.7	15.8	1.4	41.4	46.3	0.1	1.3	0.3	0.8	31.0	4.6	17.6	0.2	0.2	0.0	12.9	2.4	3.7	
	SPG [110]	17.4	0.25	45.0	28.5	0.6	0.6	64.3	49.3	0.1	0.2	0.2	0.8	48.9	27.2	24.6	0.3	2.7	0.1	20.8	15.9	0.8	
	SPLATNet [179]	18.4	0.8	64.6	39.1	0.4	0.0	58.3	58.2	0.0	0.0	0.0	0.0	71.1	9.9	19.3	0.0	0.0	0.0	23.1	5.6	0.0	
	PointNet++ [157]	20.1	6	72.0	41.8	18.7	5.6	62.3	53.7	0.9	1.9	0.2	0.2	46.5	13.8	30.0	0.9	1.0	0.0	16.9	6.0	8.9	
	TangentConv [189]	40.9	0.4	83.9	63.9	33.4	15.4	83.4	90.8	15.2	2.7	16.5	12.1	79.5	49.3	58.1	23.0	28.4	8.1	49.0	35.8	28.5	
	LatticeNet [163]	52.2	-	88.8	73.8	64.6	25.6	86.9	88.6	43.3	12.0	20.8	24.8	76.4	57.9	54.7	34.2	39.9	60.9	55.2	41.5	42.7	
	PolarNet [250]	54.3	14	90.8	74.4	61.7	21.7	90.0	93.8	22.9	40.2	30.1	28.5	84.0	65.5	67.8	43.2	40.2	5.6	61.3	51.8	57.5	
	RandLA-Net [90]	55.9	1.24	90.5	74.0	61.8	24.5	89.7	94.2	43.9	47.4	32.2	39.1	83.8	63.6	68.6	48.4	47.4	9.4	60.4	51.0	50.7	
	SqueezeSeg [218]	29.5	1	85.4	54.3	26.9	4.5	57.4	68.8	3.3	16.0	4.1	3.6	60.0	24.3	53.7	12.9	13.1	0.9	29.0	17.5	24.5	
	SqueezeSegV2 [219]	39.7	1	88.6	67.6	45.8	17.7	73.7	81.8	13.4	18.5	17.9	14.0	71.8	35.8	60.2	20.1	25.1	3.9	41.1	20.2	36.3	
	DarkNet21Seg [28]	47.4	25	91.4	74.0	57.0	26.4	81.9	85.4	18.6	26.2	26.5	15.6	77.6	48.4	63.6	31.8	33.6	4.0	52.3	36.0	50.0	
	DarkNet53Seg [28]	49.9	50	91.8	74.6	64.8	27.9	84.1	86.4	25.5	24.5	32.7	22.6	78.3	50.1	64.0	36.2	33.6	4.7	55.0	38.9	52.2	
	RangeNet53++ [144]	52.2	50	91.8	75.2	65.0	27.8	87.4	91.4	25.7	34.4	23.0	80.5	55.1	64.6	38.3	38.8	4.8	58.6	47.9	55.9		
	SalsaNext [54]	54.5	6.73	90.9	74.0	58.1	27.8	87.9	90.9	21.7	36.4	29.5	19.9	81.8	61.7	66.3	52.0	52.7	16.0	58.2	51.7	58.0	
	SqueezeSegV3 [230]	55.9	26	91.7	74.8	63.4	26.4	89.0	92.5	29.6	38.7	36.5	33.0	82.0	58.7	65.4	45.6	46.2	20.1	59.4	49.6	58.9	
Weak supervision	Ours (0.1%)	50.8	1.05	90.5	72.9	56.8	19.1	84.8	92.1	36.7	39.3	30.1	26.0	80.8	59.1	67.0	36.4	25.3	7.2	53.3	44.5	44.0	
	Ours (0.01%)	39.1	1.05	86.6	66.4	43.0	16.9	80.0	85.5	12.9	4.0	1.4	18.4	72.7	49.6	58.8	16.9	22.3	4.3	42.3	31.7	16.6	

5.4.5.1 Variants of Semantic Queries

The hierarchical point feature query mechanism is the major component of our SQN. To evaluate this component, we perform semantic queries at different encoding layers. In particular, we train four additional models, each of which has a different combination of queried neighboring point features.

Table 5.10: Ablations of different levels of semantic query.

Model	1st	2nd	3rd	4st	OA(%)	mIoU(%)
A	✓				48.66	22.89
B				✓	75.54	46.02
C	✓	✓			70.76	38.18
D	✓	✓	✓		82.37	54.21
E	✓	✓	✓	✓	86.26	61.37

From Table 5.10 we can see that the segmentation performance drops significantly if we only collect the relevant point features at a single layer (*e.g.*, the first or the last layer), whilst querying at the last layer can achieve much better results than in the first layer. This is because the points in the last encoding layer are quite sparse but representative, aggregating a large number of neighboring points. Additionally, querying at different encoding layers and combining them is likely to achieve better segmentation results, mainly because it integrates different spatial levels of semantic content and considers more neighboring points.

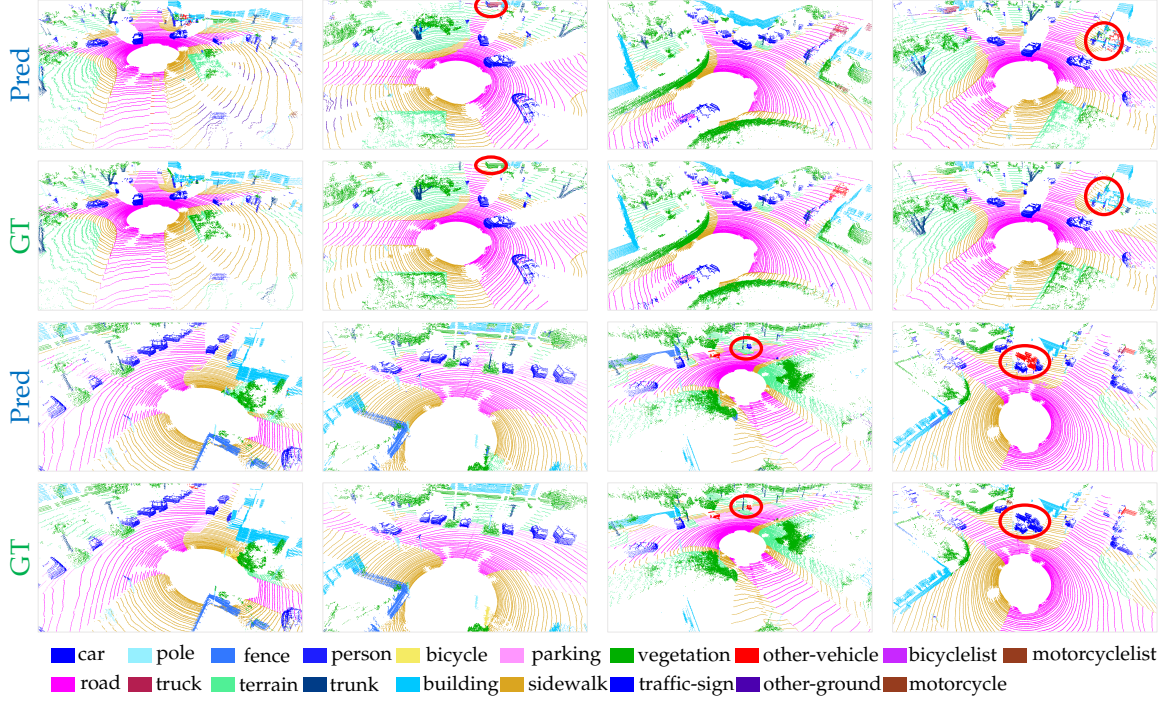


Figure 5.7: Qualitative results achieved by our SQN on the validation set (Sequence 08) of SemanticKITTI [28] dataset. The red circle highlights the failure case.

5.4.5.2 Varying Number of Queried Neighbours

Intuitively, querying a larger neighborhood is more likely to achieve better results. However, an overly large neighborhood may include points with very different semantics, diminishing overall performance. To investigate the impact of the number of neighboring points used in our semantic query, we conduct experiments by varying the number of neighboring points from 1 to 25. As shown in Figure 5.8, the overall performance with differing numbers of neighboring points does not change significantly, showing that our simple query mechanism is robust³ to the size of the neighboring patch. Instead, the mixture of different feature levels plays a more important role as demonstrated in Table 5.10.

³To avoid ambiguity, unless otherwise specified, the robust/robustness used in this thesis only means insensitivity to variations (*e.g.*, different sizes of the neighboring patch, different randomly annotated points), which is different from the “adversarial robustness” or “robustness verification”. The latter focus on providing provable guarantees for the evaluation of the robustness of the neural networks on adversarial examples [95, 220].

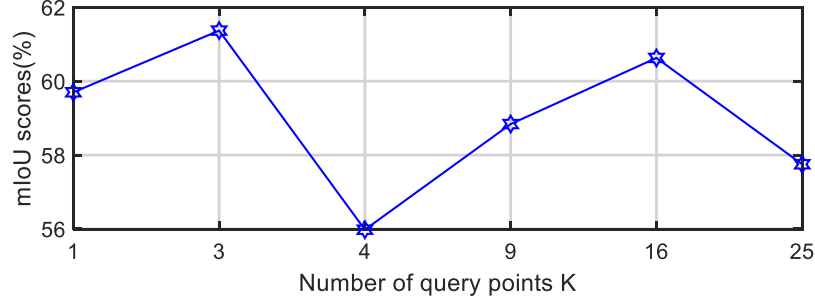


Figure 5.8: The results of our SQN with different number of querying points on the *Area-5* of the S3DIS dataset.

Table 5.11: Sensitivity analysis of the proposed SQN on the S3DIS dataset (*Area 5*) by running 5 times. Overall Accuracy (OA, %), mean IoU (mIoU, %), and per-class IoU (%) are reported. Bold represents the best result.

	OA(%)	mIoU(%)	ceil.	floor	wall	beam	col.	wind.	door	table	chair	sofa	book.	board	clut.
Iter1	86.53	60.97	92.33	96.70	78.99	0.00	25.01	56.76	58.99	74.22	79.06	58.41	67.73	53.29	51.08
Iter2	85.63	59.24	91.72	97.01	77.35	0.00	20.10	53.55	65.28	71.63	83.61	51.44	65.57	43.37	49.49
Iter3	86.39	60.93	91.96	96.02	78.88	0.00	25.31	55.80	63.43	70.71	82.80	51.18	68.39	56.53	51.05
Iter4	86.32	59.40	92.22	96.07	78.85	0.00	19.00	50.10	65.19	68.37	83.27	49.79	67.09	51.33	50.89
Iter5	86.40	61.56	91.88	95.97	78.89	0.00	24.95	55.88	63.73	70.75	83.20	59.29	68.25	56.37	51.13
Average	86.25	60.42	92.02	96.35	78.59	0.00	22.87	54.42	63.32	71.14	82.39	54.02	67.41	52.18	50.73
STD	0.32	0.93	0.22	0.42	0.62	0.00	2.74	2.41	2.29	1.88	1.68	3.99	1.03	4.82	0.62

5.4.5.3 Varying Annotated Points

To verify the sensitivity of our SQN to different randomly annotated points, we train our models five times with exactly the same architectures, *i.e.*, the only change is that different subsets of randomly selected 0.1% of points are labeled. The experimental results are reported in Table 5.11. It can be seen that there are slight, but not significant, differences between different runs. This indicates that the proposed SQN is robust to the choice of randomly annotated points. We also notice that the major performance change lies in minor categories such as *door*, *sofa*, and *board*, showing that the underrepresented classes are more sensitive to weak annotation.

5.4.5.4 Varying Proportion of Annotated Points

We further examine the performance of SQN with differing amounts of annotated points. As shown in Table 5.12, the proposed SQN can achieve satisfactory segmentation performance when there are only 0.1% labels available, but the performance drops significantly when there are only 0.01% labeled points available, primarily because the supervision signal is too sparse and limited in this case. It

is also interesting to see that our framework achieves slightly better mIoU performance when using 10% labels compared with full supervision. In particular, the performance in minority categories such as *column/window/door* has improved by 2%-5%. This implies that: 1) In a sense, the supervision signal is sufficient in this case; 2) Another way to address the critical issue of imbalanced class distribution may be to use a portion of training data (*i.e.*, weak supervision). This is an interesting direction for further research, and we leave it for future exploration.

Table 5.12: Quantitative results achieved by our SQN on *Area-5* of S3DIS under different amounts of labeled points.

Settings	Methods	mIoU(%)	<i>ceil.</i>	<i>floor</i>	<i>wall</i>	<i>beam</i>	<i>col.</i>	<i>win.</i>	<i>door</i>	<i>chair</i>	<i>table</i>	<i>book.</i>	<i>sofa</i>	<i>board</i>	<i>clutter</i>
100%	SQN	63.73	92.76	96.92	81.84	0.00	25.93	50.53	65.88	79.52	85.31	55.66	72.51	65.78	55.85
10%	SQN	64.67	93.04	97.45	81.55	0.00	28.01	55.77	68.68	80.11	87.67	55.25	72.31	63.91	57.02
1%	SQN	63.65	92.03	96.41	81.32	0.00	21.42	53.71	73.17	77.80	85.95	56.72	69.91	66.57	52.49
0.1%	SQN	61.41	91.72	95.63	78.71	0.00	24.23	55.89	63.14	70.50	83.13	60.67	67.82	56.14	50.63
0.01%	SQN	45.30	89.16	93.49	71.28	0.00	4.14	34.67	41.02	54.88	66.85	25.68	55.37	12.80	39.57

5.4.5.5 Varying Backbones of our SQN Framework

To further study the performance of our SQN framework with different backbones, we further implement our SQN based on the representative voxel-based baseline MinkowskiNet [51]. Specifically, we follow the implementation provided in [186], and the point local feature extractor, in this case, includes 4 encoding layers, each containing a 3D convolution block (kernel size and stride are set as 2) and 2 residual blocks (kernel size and stride are set as 3 and 1, respectively). Additionally, the feature query network gathers feature vectors from multi-level feature volumes through trilinear interpolation, and then simply concatenated and sent to MLPs (256-128-96) for semantic prediction.

The experimental results achieved by our SQN and baseline networks on the SemanticKITTI dataset under different weak supervision settings are shown in Table 5.14. We can see that our SQN achieves comparable performance with the baseline under 0.1% settings, primarily because the supervision signal is still sufficient at this time, considering the large scale of the dataset. However, we can clearly observe that our SQN outperforms the baseline by a large margin (6.8% improvement in mIoU scores) when there are only 0.01% points are annotated. This further demonstrates the effectiveness of our semantic query framework.

Table 5.13: Quantitative results achieved by different methods on the region-wise labeled S3DIS dataset.

	SPVCNN [186]	MinkowskiUnet [51]	SQN (Ours)
Random	49.61	46.15	60.19
Softmax Confidence [204]	51.05	45.45	57.24
Softmax Margin [204]	50.80	44.33	57.94
Softmax Entropy [204]	50.35	49.99	57.98
MC Dropout [67]	50.39	49.94	58.30
ReDAL [221]	50.89	47.88	54.24

Table 5.14: Quantitative results achieved by our SQN (SPVNAS) on the validation set of the SemanticKITTI dataset under different weak supervision settings.

Methods	mIoU(%)	Params(M)	road	sidewalk	parking	other-ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic-sign
MinkUNet 0.1%	55.5	21.9	92.5	79.0	43.0	0.9	88.7	95.0	64.5	0.9	47.4	46.4	87.3	63.4	73.7	45.3	70.3	0.3	53.4	59.9	43.2
SQN (MinkUNet) 0.1%	55.8	8.8	91.5	78.0	41.1	0.9	88.5	94.9	66.8	5.7	43.2	43.6	88.2	64.0	75.5	49.5	66.3	0.0	55.9	61.1	45.2
MinkUNet 0.01%	43.2	21.9	89.3	74.8	32.1	0.0	87.6	92.4	25.8	0.0	24.8	20.1	87.1	56.4	73.2	9.6	15.9	0.0	55.1	48.2	28.3
SQN (MinkUNet) 0.01%	50.0	8.8	89.7	75.6	31.9	0.2	87.6	93.5	47.2	0.2	35.6	31.6	88.2	58.0	76.0	33.8	59.1	0.0	52.9	52.1	36.4

5.4.5.6 Comparison with Other Feature Propagation Layers

Here, we explicitly discuss the key differences between our SQN and other general feature propagation layers used in [157, 90]. To clarify, general feature propagation layers are primarily used to recover the full spatial resolution for dense segmentation given **full supervision**, while SQN queries parallelly and hierarchically at multiple spatial resolutions, aiming to effectively propagate the limited signals to a much wider context given **weak supervision**. Further, we compare our SQN with two general feature propagation layers by conducting ablative experiments on three datasets. Specifically, we keep the feature encoder, labeling ratio, and experimental settings unchanged, and only replace the semantic query decoder as the vanilla feature propagation layers used in PointNet++ and RandLA-Net. The experimental results are shown in Table 5.15. It can be seen that our SQN achieves consistently better results compared with two general feature propagation layers, primarily due to the usage of parallelly and hierarchically semantic queries in different encoding layers, enabling the limited and sparse supervision signal to be back-propagated to a much wider context. By contrast, both the feature propagation layers used in RandLA-Net and PointNet++ can only propagate the sparse label layer by layer, with relatively limited receptive fields, hence achieving inferior performance in weak supervision settings.

Table 5.15: Ablative experiments on different feature propagation (FP) layers.

	S3DIS	ScanNet (val)	Toronto3D
FP layers in RandLA-Net [90]	52.9	51.2	71.7
FP layers in PointNet++ [157]	53.4	52.7	72.4
SQN (Ours)	61.4	58.4	77.7

5.5 Discussions

Extension to Region-wise Annotated Data. Beyond evaluating our method on the randomly point-wise annotated datasets, we also extend our SQN on the region-wise sparsely labeled S3DIS dataset. Following [221], point clouds are firstly grouped into regions by unsupervised over-segmentation methods [153], and then a sparse number of regions are manually annotated through various active learning strategies [204, 221, 67]. As shown in Table 5.13, our SQN can consistently achieve better results than vanilla SPVCNN [186] and MinkowskiNet [51] under the same supervision signal (10 iterations of active selection), regardless of the active learning strategy used. This further validates the superiority of our method under weak supervision.

Performance on Boundary Areas. We further evaluate the segmentation performance of our SQN on the boundary points, since the assumption about the consistency of neighborhood semantics may not hold in the boundary areas with different semantics. Specifically, we first define the boundary points as follows: if the queried spherical neighboring points within a radius r have different semantics, the query point is regarded as on the boundary (red points in the Figure 5.9). We then set different radii r according to the density of the point cloud, by increasing r from 0.05m to 0.2m. Not surprisingly, given a smaller r , the performance drops significantly for both RandLA-Net (full supervision) and SQN (0.1%), showing that it is still a common issue for existing methods. We will leave this issue for future exploration.

Flexibility of SQN. Thanks to the flexibility of the SQN framework, the proposed method should be able to take any point in the space as input and infer its semantic label through query and interpolation (even if that point itself does not exist in the point clouds). To further validate this, we tried to train our SQN on the partial point clouds (*i.e.*, raw point clouds), but tested on the aggregated point clouds based on the SemanticKITTI dataset. We begin by training SQN using data consisting of individual sparse LiDAR scans. Next, we combine multiple sparse Li-



	Boundary ($r=0.05$)	Boundary ($r=0.1$)	Boundary ($r=0.2$)	All Points
RandLA (Full sup.)	38.3	46.6	53.7	63.0
SQN (0.1%)	34.3	42.7	50.5	61.4

Figure 5.9: Quantitative comparison of RandLA-Net and the proposed SQN on the boundary areas.

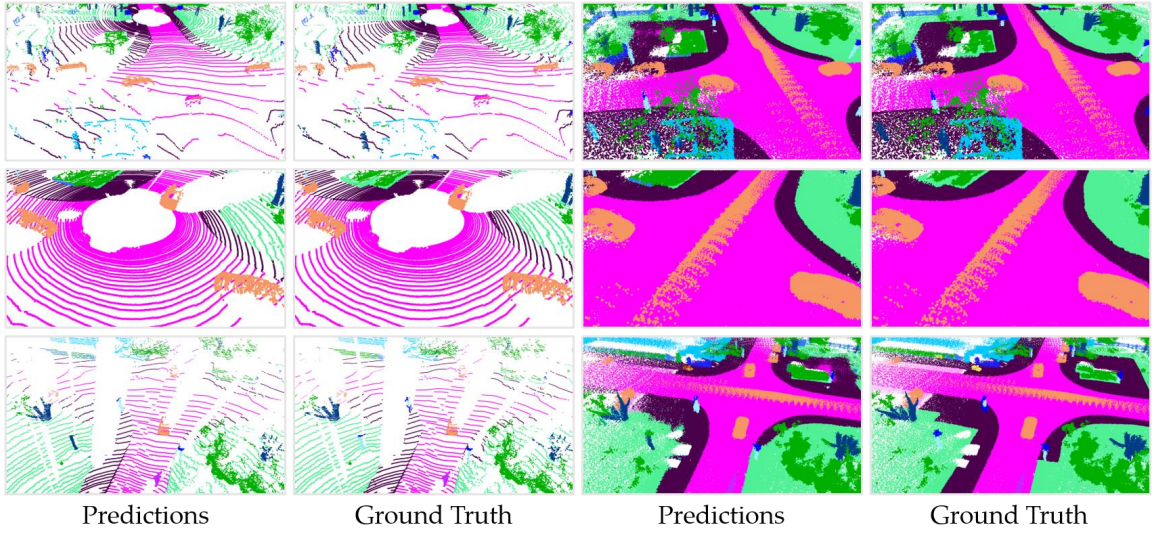


Figure 5.10: Qualitative performance achieved by SQN on the SemanticKITTI dataset when trained on partial point clouds.

DAR scans based on the provided pose information. Finally, we apply the trained SQN model to the merged point clouds. It is worth mentioning that the density of trained and tested point clouds varies significantly. The qualitative results are shown in Figure 5.10. It can be seen that the proposed SQN can still achieve satisfactory performance on the complete point clouds, even though our model is only trained with partial and incomplete point clouds. Considering that point clouds are irregularly sampled points from the continuous surface, it would be interesting to further explore the continuous semantic surface learning based on our framework.

Limitation and Future Work. Our SQN is intuitive simple yet effective. Extensive experiments have demonstrated the superiority of our method on large-scale

datasets. However, it still relies on human annotations, albeit extremely sparse. Ideally, the 3D semantics can be automatically discovered from raw point clouds. We leave this unsupervised learning of 3D semantic segmentation for our future exploration.

5.6 Summary

In this chapter, we propose SQN, a conceptually simple, elegant, and novel framework to learn the semantics of large-scale point clouds, with only 0.1% labels for training. We first point out the redundancy of dense 3D annotations through extensive experiments, and then propose an effective semantic query framework based on the assumption of semantic similarity of neighboring points in 3D space. The proposed SQN shows great potential for weakly-supervised semantic segmentation of large-scale point clouds. It would be interesting to extend this method for weakly-supervised instance segmentation, panoptic segmentation, and interactive annotation based on active learning.



Figure 5.11: Qualitative results achieved by our SQN on the validation set (Sequence 08) of SensatUrban[88] dataset. Best viewed in color.

Chapter 6

Conclusion and Future Work

6.1 Summary of Key Contributions

The overarching purpose of this thesis has been to build intelligent systems that understand the geometric structure and semantics of the 3D real-world environments. To achieve this goal, we build an urban-scale 3D point cloud benchmark and proposed solutions for semantic understanding of large-scale 3D point clouds from full supervision to weak supervision. To summarize, the key contributions of this thesis are as follows:

In Chapter 3, we introduce an urban-scale photogrammetric point cloud dataset, termed SensatUrban, as a key step towards the semantic understanding of urban 3D scenes. The SensatUrban dataset is currently the largest public-available photogrammetric point cloud dataset in terms of spatial scale and semantic annotations. Specifically, the proposed dataset consists of large areas from three UK cities, covering about 7.6 km^2 of the city landscape, with nearly three billion richly annotated points. In the dataset, each 3D point is labeled as one of 13 semantic classes. Additionally, we extensively evaluate the performance of state-of-the-art algorithms on this dataset and provide a comprehensive analysis of the results. In particular, we identify several key challenges toward urban-scale point cloud understanding, such as data preparation, imbalanced class distribution, cross-city generalization, cross-dataset generalization, and weakly (self) supervised learning *etc.* This is the first study to systematically examine the unique difficulties faced in the semantic understanding of large-scale 3D point clouds, and it offers insightful analysis and meaningful directions for further research.

In Chapter 4, we present an efficient and lightweight neural architecture RandLA-Net, to achieve semantic segmentation of large-scale unstructured 3D point clouds.

To accomplish this, we first analyzed the advantages and disadvantages of existing downsampling strategies, and identified that random sampling is a suitable component for large-scale point clouds. We then proposed a local feature aggregation module that increases the receptive field of each point and adaptively aggregates prominent features, thereby preserving overall geometric details even with aggressive random downsampling. With random downsampling and local feature aggregation as its building blocks, the proposed RandLA-Net can infer large-scale point clouds with millions of points in a single pass, with state-of-the-art accuracy and up to $200\times$ speed up compared with other similar techniques [110].

In Chapter 5, we propose a novel weakly-supervised approach called SQN, for achieving semantic segmentation of large-scale 3D point clouds. We first conducted a pilot study to investigate the necessity of dense 3D labels for learning meaningful semantic representations and found that sparse annotation with only 0.1% of points can be a critical point for weak supervision. Motivated by this, we proposed SQN to implicitly augment the available supervision signals by leveraging the semantic similarity between neighboring points. This is achieved through a neighboring point query at different latent feature maps, which allows sparse and valuable supervision signals to be propagated to a much wider context. As a result, SQN reduces the amount of required annotation to 0.1%, a significant decrease compared to 10% [234, 188] and 1% used in [249]. Additionally, we also developed a dedicated tool for sparse annotation using off-the-shelf software, which can save up to 98% of the annotation cost for large-scale 3D point clouds.

Apart from the technical contributions outlined above, I have also made efforts to contribute to the open-source community. Specifically, the datasets and algorithms proposed in this thesis have been made available on GitHub, the repositories containing our implementation can be accessed via the following links:

- The SensatUrban dataset (Chapter 3) can be accessed through this link¹. It has been downloaded over 1,000 times and we have also hosted two Urban3D challenges based on this dataset.
- Our implementation of RandLA-Net (Chapter 4) is available at this GitHub repository². It has received over 1,000 stars to date and is also included in the Open3D and TorchPoints3D libraries.
- Our implementation of SQN (Chapter 5) is available at this repository³.

¹<https://github.com/QingyongHu/SensatUrban>

²<https://github.com/QingyongHu/RandLA-Net>

³<https://github.com/QingyongHu/SQN>

6.2 Future Work

Despite this thesis having made contributions in advancing the semantic understanding of large-scale 3D scenes, it is important to note that this field is rapidly evolving and there are numerous promising technologies and avenues worthy of further research. Here I list a few examples of potential future works as follows.

Unsupervised semantic clustering of 3D point clouds. This is based on the fact that the massive unlabelled 3D point clouds have not been fully exploited. It is noted that a single vehicle-mounted LiDAR captures up to 84 billion points per day, while the number of points for existing largest annotated datasets is only 4 billion. Considering the extremely large quantity of unlabeled data, it has great potential to further investigate how to effectively learn from the unstructured, complex, and unlabeled 3D data. In particular, we aim to group data points into semantically homogeneous classes through unsupervised clustering. Point-wise feature similarity and geometric consistency after transformation will be further investigated, as well as the contrastive learning paradigm, are expected to be explored in future work.

Robust and explainable algorithms. Although the semantic understanding of 3D surroundings has made significant progress in terms of efficiency and accuracy in recent years, existing neural networks will still inevitably produce some unexpected outputs (*e.g.*, ridiculous misclassification). In addition, most of the existing neural network does not consider the security and robustness issues in the design process, hence usually vulnerable to adversarial attacks. All these issues prevent these techniques from being applied to safety-critical applications such as autonomous driving and medical-related analysis. Therefore, how to enhance the interpretability of neural network predictions and provide reliable outputs is a direction worthy of further research.

Unified and transferable 3D representation. Considering the distinct data collection mechanisms of different LiDAR sensors (Velodyne, Hesai, Livox, Robosense), the density, data distribution, and noise pattern of acquired point clouds are usually significantly different. In this case, it is difficult to transfer and generalize a model trained on one type of point cloud to another mode of data. Therefore, it is necessary and desirable to explore unified and transferable 3D representations. In light of the representation in our 3D physical world being continuous and singular, we aim to develop a general 3D representation, which is likely to be continuous and can act as an intermediate representation to transfer across a

variety of different modalities. As such, the domain discrepancies induced by different LiDAR sensors can be overcome, and the trained model can be generalized to different data modalities.

Bibliography

- [1] 3dtk-the 3d toolkit. <https://slam6d.sourceforge.io/images/bremenCity.jpg>. Accessed: 2023-03-21.
- [2] 64-channel lidar sensor hdl-64e. https://img.directindustry.com/images_d/i/photo-g/182407-10533478.webp. Accessed: 2023-03-21.
- [3] Bumblebee2. <https://voltrium.wordpress.com/machine-vision/home/stereo-products>. Accessed: 2023-03-21.
- [4] Digital scan of notre dame offers hope for its restoration after massive fire. https://www.maintworld.com/var/ezwebin_site/storage/images/media/images/notre/53656-1-eng-GB/notre.png. Accessed: 2023-03-21.
- [5] Elysium. <https://www.elysium-global.com/en/news1/>. Accessed: 2023-03-21.
- [6] Intel realsense 3d camera with lenovo. <https://www.youtube.com/watch?v=uINRC83tlTA>. Accessed: 2023-03-21.
- [7] Intel realsense lidar camera technology redefines computer vision. https://simplecore.intel.com/newsroom-de-de/wp-content/uploads/sites/21/2019/12/Intel-L515-2-960x750_c.jpg. Accessed: 2023-03-21.
- [8] Intersection over union (iou) in object detection and segmentation. <https://learnopencv.com/intersection-over-union-iou-in-object-detection-and-segmentation/>. Accessed: 2023-01-14.
- [9] Kinect wikipedia. <https://en.wikipedia.org/wiki/Kinect>. Accessed: 2023-03-21.

- [10] Lidar can cause damage to human eyes. <https://www.cleanfuture.co.in/2021/03/15/lidar-can-cause-damage-human-eyes/>. Accessed: 2023-03-21.
- [11] Lyft level 5 dataset. <https://self-driving.lyft.com/level5/data/>.
- [12] Optech ilris. <http://www.gstdubai.com/lidar/ilris.html>. Accessed: 2023-03-21.
- [13] PointNet slides. http://stanford.edu/~rqi/pointnet/docs/cvpr17_pointnet_slides.pdf.
- [14] Quanergy m8 ultra point cloud. <https://lidarnews.com/articles/quanergy-m8-ultra-digital-lidar/>. Accessed: 2023-03-21.
- [15] Sensefly ebee x fixed-wing drone. https://www.dronefly.com/media/webp_image/catalog/product/e/b/ebee-x-fixed_plates-sqr.webp. Accessed: 2023-03-21.
- [16] Sensefly s.o.d.a. 3d. <https://sensefly.aero/wp-content/uploads/2021/11/SF050016-senseFly-S.O.D.A.-3D-1-2048x1280.jpg>. Accessed: 2023-03-21.
- [17] Understanding point cloud quality perception. <https://vrtogether.eu/wp-content/uploads/2018/01/Pointcloud1.jpg>. Accessed: 2023-03-21.
- [18] Understanding portable measurement arms. <https://media.faro.com/-/media/Project/FARO/FARO/FARO/Images/Resources/2021/01/15/23/00/QuantumS-FAROArm-No-Grip-Left-Upright-FM-Product-HR-small.jpg?h=525&w=350&rev=a77ealfeae2d4fb6a32b0c884597a09f&hash=14B8ABB19B6FF504398EEF00D78DDFA0>. Accessed: 2023-03-21.
- [19] Velodyne lidar. <https://velodynelidar.com/products/puck>. Accessed: 2023-03-21.
- [20] Vivid910. https://www.researchgate.net/figure/Vivid-910-digitizer-Konica-Minolta_fig1_58715683. Accessed: 2023-03-21.

- [21] Zed camera. <https://d2t1xqejof9utc.cloudfront.net/screenshots/pics/0a8d0d9e8fb87355477c805a9e498665/large.JPG>. Accessed: 2023-03-21.
- [22] Abubakar Abid, Muhammad Fatih Balin, and James Zou. Concrete autoencoders for differentiable feature selection and reconstruction. In International Conference on Machine Learning, 2019.
- [23] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dhar-masiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9902–9912, 2022.
- [24] Eren Erdal Aksoy, Saimir Baci, and Selcuk Cavdar. Salsanet: Fast road and vehicle segmentation in LiDAR point clouds for autonomous driving. In 2020 IEEE Intelligent Vehicles Symposium (IV), pages 926–932, 2019.
- [25] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spin-net: Learning a general surface descriptor for 3D point cloud registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11753–11762, 2021.
- [26] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2D-3D-semantic data for indoor scene understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2017.
- [27] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In Asian Conference on Computer Vision, pages 180–196, 2016.
- [28] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. SemanticKITTI: A dataset for semantic scene understanding of lidar sequences. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9297–9307, 2019.
- [29] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4413–4421, 2018.

- [30] Liefeng Bo, Kevin Lai, Xiaofeng Ren, and Dieter Fox. Object recognition with hierarchical kernel descriptors. In CVPR 2011, pages 1729–1736. IEEE, 2011.
- [31] Alexandre Boulch. Generalizing discrete convolutions for unstructured point clouds. arXiv preprint arXiv:1904.02375, 2019.
- [32] Alexandre Boulch, Bertrand Le Saux, and Nicolas Audebert. Unstructured point cloud semantic labeling using deep segmentation networks. 3DOR@Eurographics, 3, 2017.
- [33] Alexandre Boulch, Gilles Puy, and Renaud Marlet. Lightconvpoint: convolution for points. In Asian Conference on Computer Vision, 2020.
- [34] Robert Bridson. Fast poisson disk sampling in arbitrary dimensions. SIGGRAPH sketches, 10(1):1, 2007.
- [35] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11621–11631, 2020.
- [36] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In 7th IEEE International Conference on 3D Vision, 3DV 2017, pages 667–676, 2018.
- [37] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012, 2015.
- [38] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3D tracking and forecasting with rich maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8748–8757, 2019.

- [39] Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang, and Liang Lin. ClusterNet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4994–5002, 2019.
- [40] Lin-Zhuo Chen, Xuan-Yi Li, Deng-Ping Fan, Ming-Ming Cheng, Kai Wang, and Shao-Ping Lu. LSANet: Feature learning on point sets by local spatial attention. arXiv preprint arXiv:1905.05442, 2019.
- [41] Meida Chen, Qingyong Hu, Thomas Hugues, Andrew Feng, Yu Hou, Kyle McCullough, and Lucio Soibelman. Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset. arXiv preprint arXiv:2203.09065, 2022.
- [42] Siheng Chen, Sufeng Niu, Tian Lan, and Baoan Liu. PCT: Large-scale 3D point cloud representations via graph inception networks with applications to autonomous driving. In IEEE international conference on image processing (ICIP), pages 4395–4399, 2019.
- [43] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In International Conference on Machine Learning, pages 1597–1607, 2020.
- [44] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1907–1915, 2017.
- [45] Ye Chen, Jinxian Liu, Bingbing Ni, Hang Wang, Jiancheng Yang, Ning Liu, Teng Li, and Qi Tian. Shape self-correction for unsupervised point cloud understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8382–8391, 2021.
- [46] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9775–9784, 2019.

- [47] Yujin Chen, Matthias Nießner, and Angela Dai. 4dcontrast: Contrastive learning with dynamic correspondences for 3d scene understanding. In European Conference on Computer Vision, pages 543–560. Springer, 2022.
- [48] Mingmei Cheng, Le Hui, Jin Xie, and Jian Yang. Sspc-net: Semi-supervised semantic 3D point cloud segmentation network. arXiv preprint arXiv:2104.07861, 2021.
- [49] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. 2-S3Net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 12547–12556, 2021.
- [50] Taying Cheng, Qingyong Hu, Qian Xie, Andrew Markham, and Niki Trigoni. Meta-sampler: Almost-universal yet task-oriented sampling for point clouds. arXiv preprint arXiv, 2022.
- [51] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
- [52] Chin Seng Chua and Ray Jarvis. Point signatures: A new representation for 3D object recognition. International Journal of Computer Vision, 25(1):63–85, 1997.
- [53] Jhonatan Contreras and Joachim Denzler. Edge-convolution point net for semantic segmentation of large-scale point clouds. In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, pages 5236–5239, 2019.
- [54] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. SalsaNext: Fast semantic segmentation of lidar point clouds for autonomous driving. arXiv preprint arXiv:2003.03653, 2020.
- [55] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5828–5839, 2017.

- [56] Mark De Deuge, Alastair Quadros, Calvin Hung, and Bertrand Douillard. Unsupervised feature learning for classification of outdoor 3D scans. In Australasian Conference on Robotics and Automation, volume 2, page 1, 2013.
- [57] Shuang Deng, Qiulei Dong, Bo Liu, and Zhanyi Hu. Superpoint-guided semi-supervised semantic segmentation of 3d point clouds. In 2022 International Conference on Robotics and Automation (ICRA), pages 9214–9220. IEEE, 2022.
- [58] Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sample. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2760–2769, 2019.
- [59] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring spatial context for 3D semantic segmentation of point clouds. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 716–724, 2017.
- [60] Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dilated point convolutions: On the receptive field of point convolutions. In British Machine Vision Conference, 2019.
- [61] Francis Engelmann, Theodora Kontogianni, Jonas Schult, and Bastian Leibe. Know what your neighbors do: 3D semantic segmentation of point clouds. In European Conference on Computer Vision, 2018.
- [62] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303–338, 2010.
- [63] Guangchi Fang, Qingyong Hu, Hanyun Wang, Yiling Xu, and Yulan Guo. 3DAC: Learning attribute compression for point clouds. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022.
- [64] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. In BMVC, 2020.

- [65] Kent Fujiwara and Taiichi Hashimoto. Neural implicit embedding for point cloud analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11734–11743, 2020.
- [66] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4340–4349, 2016.
- [67] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning, pages 1050–1059, 2016.
- [68] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:1704.06857, 2017.
- [69] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 32(11):1231–1237, 2013.
- [70] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3354–3361, 2012.
- [71] Markus Gerke and Norman Kerle. Automatic structural seismic damage assessment with airborne oblique pictometry© imagery. Photogrammetric Engineering & Remote Sensing, 77(9):885–898, 2011.
- [72] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. A2D2: Audi autonomous driving dataset. arXiv preprint arXiv:2004.06320, 2020.
- [73] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9224–9232, 2018.

- [74] Fabian Groh, Patrick Wieschollek, and Hendrik P. A. Lensch. Flex-convolution (million-scale point-cloud learning beyond grid-worlds). In Asia Conference on Computer Vision, pages 105–122, 2018.
- [75] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. IEEE transactions on pattern analysis and machine intelligence, 43(12):4338–4364, 2020.
- [76] T Hackel, N Savinov, L Ladicky, JD Wegner, K Schindler, and M Pollefeys. Semantic3d. net: A new large-scale point cloud classification benchmark. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2017.
- [77] Timo Hackel, Jan D Wegner, and Konrad Schindler. Fast semantic segmentation of 3D point clouds with strongly varying density. ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, 3:177–184, 2016.
- [78] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3D instance segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2940–2949, 2020.
- [79] A Handa, V Patraucean, V Badrinarayanan, S Stent, and R Cipolla. SceneNet: understanding real world indoor scenes with synthetic data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2016.
- [80] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: A network with an edge. ACM Transactions on Graphics (TOG), 38(4):90, 2019.
- [81] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9729–9738, 2020.
- [82] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.

- [83] Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. Monte carlo convolution for learning on non-uniformly sampled point clouds. ACM Transactions on Graphics (TOG), 37(6):1–12, 2018.
- [84] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15587–15597, 2021.
- [85] Lei Hou, Ying Wang, Xiangyu Wang, Nicoleta Maynard, Ian T Cameron, Shaohua Zhang, and Yi Jiao. Combining photogrammetry and augmented reality towards an integrated facility management system for the oil industry. Proceedings of the IEEE, 102(2):204–220, 2014.
- [86] Jinhui Hu, Suyu You, and Ulrich Neumann. Approaches to large-scale urban modeling. IEEE Computer Graphics and Applications, 23(6):62–69, 2003.
- [87] Qingyong Hu, Bo Yang, Guangchi Fang, Yulan Guo, Ales Leonardis, Niki Trigoni, and Andrew Markham. Sqn: Weakly-supervised semantic segmentation of large-scale 3D point clouds with 1000x fewer labels. arXiv preprint arXiv:2104.04891, 2021.
- [88] Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, and Andrew Markham. Towards semantic segmentation of urban-scale 3D point clouds: A dataset, benchmarks and challenges. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4977–4987, 2021.
- [89] Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, and Andrew Markham. Sensaturban: Learning semantics from urban-scale photogrammetric point clouds. International Journal of Computer Vision, pages 1–28, 2022.
- [90] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11108–11117, 2020.

- [91] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Learning semantic segmentation of large-scale point clouds with random sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.
- [92] Zeyu Hu, Xuyang Bai, Runze Zhang, Xin Wang, Guangyuan Sun, Hongbo Fu, and Chiew-Lan Tai. Lidal: Inter-frame uncertainty based active learning for 3d lidar semantic segmentation. In European Conference on Computer Vision, pages 248–265. Springer, 2022.
- [93] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 984–993, 2018.
- [94] Qianguai Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3D segmentation of point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
- [95] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In International conference on computer aided verification, pages 3–29. Springer, 2017.
- [96] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 MB model size. arXiv preprint arXiv:1602.07360, 2016.
- [97] Li Jiang, Shaoshuai Shi, Zhuotao Tian, Xin Lai, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Guided point contrastive learning for semi-supervised point cloud semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6423–6432, 2021.
- [98] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu, and Jiaya Jia. Hierarchical point-edge interaction network for point cloud semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10433–10441, 2019.
- [99] Mingyang Jiang, Yiran Wu, and Cewu Lu. PointSIFT: A sift-like network module for 3D point cloud semantic segmentation. arXiv preprint arXiv:1807.00652, 2018.

- [100] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR), 2015.
- [101] Michael Kölle, Dominik Laupheimer, Stefan Schmohl, Norbert Haala, Franz Rottensteiner, Jan Dirk Wegner, and Hugo Ledoux. H3D: Benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from uav lidar and multi-view-stereo. arXiv preprint arXiv:2102.05346, 2021.
- [102] Artem Komarichev, Zichun Zhong, and Jing Hua. A-CNN: Annularly convolutional neural networks on point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7421–7430, 2019.
- [103] Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. Lasermix for semi-supervised lidar semantic segmentation. arXiv preprint arXiv:2207.00026, 2022.
- [104] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In Advances in Neural Information Processing Systems, pages 109–117, 2011.
- [105] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25:1097–1105, 2012.
- [106] Hamid Laga, Yulan Guo, Hedi Tabia, Robert B Fisher, and Mohammed Benamoun. 3D Shape Analysis: Fundamentals, Theory, and Applications. John Wiley & Sons, 2018.
- [107] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In International Conference on Learning Representations (ICLR), 2017.
- [108] Shiyi Lan, Ruichi Yu, Gang Yu, and Larry S Davis. Modeling local geometric structure of 3D point clouds using Geo-CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 998–1008, 2019.

- [109] Loic Landrieu and Mohamed Boussaha. Point cloud oversegmentation with graph-structured deep metric learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7440–7449, 2019.
- [110] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4558–4567, 2018.
- [111] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12697–12705, 2019.
- [112] Truc Le and Ye Duan. PointGrid: A deep network for 3D shape understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9204–9214, 2018.
- [113] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436–444, 2015.
- [114] Huan Lei, Naveed Akhtar, and Ajmal Mian. Octree guided cnn with spherical kernels for 3D point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
- [115] Huan Lei, Naveed Akhtar, and Ajmal Mian. SegGCN: Efficient 3D point cloud segmentation with fuzzy spherical kernel. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
- [116] Huan Lei, Naveed Akhtar, and Ajmal Mian. Spherical kernel for efficient graph convolution on 3D point clouds. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.
- [117] Huan Lei, Naveed Akhtar, and Ajmal Mian. Spherical kernel for efficient graph convolution on 3D point clouds. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.
- [118] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3D lidar using fully convolutional network. In Robotics: Science and Systems, 2016.

- [119] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. DeepGCNs: Can gcns go as deep as cnns? In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9267–9276, 2019.
- [120] Jiaxin Li, Ben M Chen, and Gim Hee Lee. SO-Net: Self-organizing network for point cloud analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9397–9406, 2018.
- [121] Xinke Li, Chongshou Li, Zekun Tong, Andrew Lim, Junsong Yuan, Yuwei Wu, Jing Tang, and Raymond Huang. Campus3D: A photogrammetry point cloud benchmark for hierarchical understanding of outdoor scene. In Proceedings of the 28th ACM International Conference on Multimedia, pages 238–246, 2020.
- [122] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on X-transformed points. In Advances in Neural Information Processing Systems, volume 31, 2018.
- [123] Ying Li, Lingfei Ma, Zilong Zhong, Dongpu Cao, and Jonathan Li. Tgnet: Geometric graph cnn on 3-d point cloud segmentation. IEEE Transactions on Geoscience and Remote Sensing, 58(5):3588–3600, 2019.
- [124] Zhidong Liang, Ming Yang, Liuyuan Deng, Chunxiang Wang, and Bing Wang. Hierarchical depthwise graph convolutional neural network for 3D semantic segmentation of point clouds. In International Conference on Robotics and Automation (ICRA), pages 8152–8158, 2019.
- [125] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2980–2988, 2017.
- [126] Yaping Lin, George Vosselman, and Michael Ying Yang. Weakly supervised semantic segmentation of airborne laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 187:79–100, 2022.
- [127] Fangyu Liu, Shuaipeng Li, Liqiang Zhang, Chenghu Zhou, Rongtian Ye, Yuebin Wang, and Jiwen Lu. 3DCNN-DQN-RNN: A deep reinforcement learning framework for semantic parsing of large-scale 3D point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5678–5687, 2017.

- [128] Jinxian Liu, Bingbing Ni, Caiyuan Li, Jiancheng Yang, and Qi Tian. Dynamic points agglomeration for hierarchical point sets learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 7546–7555, 2019.
- [129] Minghua Liu, Yin Zhou, Charles R Qi, Boqing Gong, Hao Su, and Dragomir Anguelov. Less: Label-efficient semantic segmentation for lidar point clouds. In European Conference on Computer Vision, pages 70–89. Springer, 2022.
- [130] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. MeteorNet: Deep learning on dynamic 3D point cloud sequences. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9246–9255, 2019.
- [131] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8895–8904, 2019.
- [132] Yunze Liu, Li Yi, Shanghang Zhang, Qingnan Fan, Thomas Funkhouser, and Hao Dong. P4contrast: Contrastive learning with pairs of point-pixel pairs for rgb-d scene understanding. arXiv preprint arXiv:2012.13089, 2020.
- [133] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. 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, pages 1726–1736, 2021.
- [134] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3D deep learning. In Advances in Neural Information Processing Systems, 2019.
- [135] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3431–3440, 2015.
- [136] Haifeng Luo, Chongcheng Chen, Lina Fang, Kourosh Khoshelham, and Guixi Shen. MS-RRFSegNet: Multiscale regional relation feature segmentation network for semantic segmentation of urban scene point clouds. IEEE Transactions on Geoscience and Remote Sensing, 58(12):8301–8315, 2020.

- [137] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3569–3577, 2018.
- [138] Lingfei Ma, Ying Li, Jonathan Li, Weikai Tan, Yongtao Yu, and Michael A Chapman. Multi-scale point-wise convolutional neural networks for 3D object segmentation from lidar point clouds in large-scale environments. IEEE Transactions on Intelligent Transportation Systems, 22(2):821–836, 2019.
- [139] Yanni Ma, Yulan Guo, Hao Liu, Yinjie Lei, and Gongjian Wen. Global context reasoning for semantic segmentation of 3D point clouds. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2931–2940, 2020.
- [140] Jiageng Mao, Xiaogang Wang, and Hongsheng Li. Interpolated convolutional networks for 3D point cloud understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1578–1587, 2019.
- [141] Ken Martin. Using virtual reality devices with vtk. <https://www.kitware.com/using-virtual-reality-devices-with-vtk/>. Accessed: 2023-03-21.
- [142] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. SceneNet RGB-D: 5m photorealistic images of synthetic indoor trajectories with ground truth. arXiv preprint arXiv:1612.05079, 2016.
- [143] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. Vv-net: Voxel vae net with group convolutions for point cloud segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8500–8508, 2019.
- [144] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. RangeNet++: Fast and accurate lidar semantic segmentation. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4213–4220, 2019.

- [145] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In International Conference on Machine Learning, pages 1791–1799, 2014.
- [146] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 909–918, 2019.
- [147] Javier A Montoya-Zegarra, Jan D Wegner, L’ubor Ladický, and Konrad Schindler. Mind the gap: modeling local and global context in (road) networks. In German Conference on Pattern Recognition, pages 212–223, 2014.
- [148] Michael E Mortenson. Geometric modeling. John Wiley & Sons, Inc., 1997.
- [149] Daniel Munoz, J Andrew Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual classification with functional max-margin markov networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 975–982, 2009.
- [150] Muzammal Naseer, Salman Khan, and Fatih Porikli. Indoor scene understanding in 2.5/3d for autonomous agents: A survey. IEEE access, 7:1859–1887, 2018.
- [151] Anshul Paigwar, Ozgur Erkent, Christian Wolf, and Christian Laugier. Attentional pointnet for 3D-object detection in point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 0–0, 2019.
- [152] Yancheng Pan, Biao Gao, Jilin Mei, Sibao Geng, Chengkun Li, and Huijing Zhao. Semanticpos: A point cloud dataset with large quantity of dynamic instances. arXiv preprint arXiv:2002.09147, 2020.
- [153] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2027–2034, 2013.

- [154] Omid Poursaeed, Tianxing Jiang, Quintessa Qiao, Nayun Xu, and Vladimir G Kim. Self-supervised learning of point clouds via orientation estimation. arXiv preprint arXiv:2008.00305, 2020.
- [155] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 652–660, 2017.
- [156] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5648–5656, 2016.
- [157] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In Advances in Neural Information Processing Systems, 2017.
- [158] Nannan Qin, Weikai Tan, Lingfei Ma, Dedong Zhang, and Jonathan Li. Opengf: An ultra-large-scale ground filtering dataset built upon open als point clouds around the world. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1082–1091, 2021.
- [159] Deepak Rao, Quoc V Le, Thanathorn Phoka, Morgan Quigley, Attawith Sudsang, and Andrew Y Ng. Grasping novel objects with depth segmentation. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2578–2585, 2010.
- [160] Dario Rethage, Johanna Wald, Jurgen Sturm, Nassir Navab, and Federico Tombari. Fully-convolutional point networks for large-scale point clouds. In European Conference on Computer Vision, pages 596–611, 2018.
- [161] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3D representations at high resolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3577–3586, 2017.
- [162] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The SYNTHIA dataset: A large collection of synthetic images

- for semantic segmentation of urban scenes. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3234–3243, 2016.
- [163] Radu Alexandru Rosu, Peer Schütt, Jan Quenzel, and Sven Behnke. LatticeNet: Fast point cloud segmentation using permutohedral lattices. arXiv preprint arXiv:1912.05905, 2019.
- [164] Franz Rottensteiner, Gunho Sohn, Jaewook Jung, Markus Gerke, Caroline Baillard, Sebastien Benitez, and Uwe Breitenkopf. The ISPRS benchmark on urban object classification and 3D building reconstruction. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1, 1(1):293–298, 2012.
- [165] Xavier Roynard, Jean-Emmanuel Deschaud, and François Goulette. Classification of point cloud for road scene understanding with multiscale voxel deep network. In 10th workshop on Planning, Perception and Navigation for Intelligent Vehicles PPNIV’2018, 2018.
- [166] Xavier Roynard, Jean-Emmanuel Deschaud, and François Goulette. Classification of point cloud scenes with multiscale voxel deep network. arXiv preprint arXiv:1804.03583, 2018.
- [167] Xavier Roynard, Jean-Emmanuel Deschaud, and François Goulette. Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. The International Journal of Robotics Research, 37(6):545–557, 2018.
- [168] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3D registration. In International Conference on Robotics and Automation (ICRA), pages 3212–3217, 2009.
- [169] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. Advances in Neural Information Processing Systems, pages 12962–12972, 2019.
- [170] Andrés Serna, Beatriz Marcotegui, François Goulette, and Jean-Emmanuel Deschaud. Paris-rue-madame database: a 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In 4th International Conference on Pattern Recognition, Applications and Methods ICPRAM 2014, 2014.

- [171] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
- [172] Xian Shi, Xun Xu, Ke Chen, Lile Cai, Chuan Sheng Foo, and Kui Jia. Label-efficient point cloud semantic segmentation: An active learning approach. arXiv preprint arXiv:2101.06931, 2021.
- [173] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In European conference on computer vision, pages 746–760, 2012.
- [174] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3693–3702, 2017.
- [175] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun RGB-D: A RGB-D scene understanding benchmark suite. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 567–576, 2015.
- [176] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun RGB-D: A RGB-D scene understanding benchmark suite. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 567–576, 2015.
- [177] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1746–1754, 2017.
- [178] Alex Starovoitau. Visualizing lidar data. <https://navoshta.com/kit-ti-lidar/>. Accessed: 2023-03-21.
- [179] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: sparse lattice networks for point cloud processing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2530–2539, 2018.

- [180] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In Proceedings of the IEEE international conference on computer vision, pages 2686–2694, 2015.
- [181] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2446–2454, 2020.
- [182] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey Hinton, and Kwang Moo Yi. Canonical capsules: Unsupervised capsules in canonical pose. arXiv preprint arXiv:2012.04718, 2020.
- [183] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. Advances in neural information processing systems, 12, 1999.
- [184] Andrew Tallon. A digital scan of notre dame cathedral in paris. https://paw.princeton.edu/sites/default/files/styles/portrait_feature/public/images/content/Prince-TallonNotreDame7.jpg?itok=K53RFLgD. Accessed: 2023-03-21.
- [185] Weikai Tan, Nannan Qin, Lingfei Ma, Ying Li, Jing Du, Guorong Cai, Ke Yang, and Jonathan Li. Toronto-3D: A large-scale mobile lidar dataset for semantic segmentation of urban roadways. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pages 202–203, 2020.
- [186] Haotian* Tang, Zhijian* Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3D architectures with sparse point-voxel convolution. In European Conference on Computer Vision, pages 685–702, 2020.
- [187] An Tao, Yueqi Duan, Yi Wei, Jiwen Lu, and Jie Zhou. Seggroup: Seg-level supervision for 3D instance and semantic segmentation. arXiv preprint arXiv:2012.10217, 2020.

- [188] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Advances in Neural Information Processing Systems, pages 1195–1204, 2017.
- [189] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3D. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3887–3896, 2018.
- [190] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. SEGCloud: Semantic segmentation of 3D point clouds. In International Conference on 3D Vision (3DV), pages 537–547, 2017.
- [191] Ali Thabet, Humam Alwassel, and Bernard Ghanem. Self-supervised learning of local features in 3D point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 938–939, 2020.
- [192] Hugues Thomas, François Goulette, Jean-Emmanuel Deschaud, and Beatriz Marcotegui. Semantic classification of 3D point clouds with multiscale spherical neighborhoods. In International Conference on 3D Vision (3DV), pages 390–398, 2018.
- [193] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KPConv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6411–6420, 2019.
- [194] Guofeng Tong, Yong Li, Dong Chen, Qi Sun, Wei Cao, and Guiqiu Xiang. Cspc-dataset: New lidar point cloud dataset and benchmark for large-scale scene semantic segmentation. IEEE access, 8:87695–87718, 2020.
- [195] Giang Truong, Syed Zulqarnain Gilani, Syed Mohammed Shamsul Islam, and David Suter. Fast point cloud registration using semantic segmentation. In 2019 Digital Image Computing: Techniques and Applications (DICTA), pages 1–8, 2019.

- [196] Benjamin Ummerhofer, Lukas Prantl, Nils Thuerey, and Vladlen Koltun. Lagrangian fluid simulation with continuous convolutions. In International Conference on Learning Representations (ICLR), 2019.
- [197] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1588–1597, 2019.
- [198] Abhinav Valada, Johan Vertens, Ankit Dhall, and Wolfram Burgard. Adap-net: Adaptive semantic segmentation in adverse environmental conditions. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 4644–4651, 2017.
- [199] Bruno Vallet, Mathieu Brédif, Andrés Serna, Beatriz Marcotegui, and Nicolas Paparoditis. Terramobilita/iqmulus urban point cloud analysis benchmark. Computers & Graphics, 49:126–133, 2015.
- [200] Nina Varney, Vijayan K Asari, and Quinn Graehling. Dales: a large-scale aerial lidar data set for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 186–187, 2020.
- [201] Nina Varney, Vijayan K Asari, and Quinn Graehling. Pyramid point: A multi-level focusing network for revisiting feature layers. arXiv preprint arXiv:2011.08692, 2020.
- [202] Brian H. Wang, Wei-Lun Chao, Yan Wang, Bharath Hariharan, Kilian Q. Weinberger, and Mark Campbell. LDLS: 3-D Object Segmentation Through Label Diffusion From 2-D Images. IEEE Robotics and Automation Letters, 4(3):2902–2909, 2019.
- [203] Chu Wang, Babak Samari, and Kaleem Siddiqi. Local spectral graph convolution for point set feature learning. In European Conference on Computer Vision, pages 52–66, 2018.
- [204] Dan Wang and Yi Shang. A new active labeling method for deep learning. In 2014 International joint conference on neural networks (IJCNN), pages 112–119, 2014.

- [205] Haiyan Wang, Xuejian Rong, Liang Yang, Shuihua Wang, and Yingli Tian. Towards weakly supervised semantic segmentation in 3d graph-structured point clouds of wild scenes. In BMVC, page 284, 2019.
- [206] Haiyan Wang, Xuejian Rong, Liang Yang, Shuihua Wang, and Yingli Tian. Towards weakly supervised semantic segmentation in 3D graph-structured point clouds of wild scenes. In British Machine Vision Conference, page 284, 2019.
- [207] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9782–9792, 2021.
- [208] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10296–10305, 2019.
- [209] Puzuo Wang and Wei Yao. A new weakly supervised approach for als point cloud semantic segmentation. arXiv preprint arXiv:2110.01462, 2021.
- [210] Puzuo Wang and Wei Yao. A new weakly supervised approach for als point cloud semantic segmentation. ISPRS Journal of Photogrammetry and Remote Sensing, 188:237–254, 2022.
- [211] Renhao Wang, Marjan Albooyeh, and Siamak Ravanbakhsh. Equivariant maps for hierarchical structures. arXiv preprint arXiv:2006.03627, 2020.
- [212] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2589–2597, 2018.
- [213] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. Acm Transactions On Graphics (TOG), 38(5):1–12, 2019.
- [214] Alan Watt. 3d computer graphics. 1993.

- [215] Jiacheng Wei, Guosheng Lin, Kim-Hui Yap, Tzu-Yi Hung, and Lihua Xie. 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, pages 4384–4393, 2020.
- [216] Jiacheng Wei, Guosheng Lin, Kim-Hui Yap, Fayao Liu, and Tzu-Yi Hung. Dense supervision propagation for weakly supervised semantic segmentation on 3D point clouds. arXiv preprint arXiv:2107.11267, 2021.
- [217] Matthew J Westoby, James Brasington, Niel F Glasser, Michael J Hambrey, and Jennifer M Reynolds. ‘structure-from-motion’ photogrammetry: A low-cost, effective tool for geoscience applications. Geomorphology, 179:300–314, 2012.
- [218] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. SqueezeSeg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3D lidar point cloud. In International Conference on Robotics and Automation (ICRA), pages 1887–1893, 2018.
- [219] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In International Conference on Robotics and Automation (ICRA), pages 4376–4382, 2019.
- [220] Min Wu. Robustness evaluation of deep neural networks with provable guarantees. PhD thesis, University of Oxford, 2020.
- [221] Tsung-Han Wu, Yueh-Cheng Liu, Yu-Kai Huang, Hsin-Ying Lee, Hung-Ting Su, Ping-Chia Huang, and Winston H Hsu. Redal: Region-based and diversity-aware active learning for point cloud semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 15510–15519, 2021.
- [222] Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9621–9630, 2018.
- [223] Yushuang Wu, Zizheng Yan, Shengcai Cai, Guanbin Li, Yizhou Yu, Xiaoguang Han, and Shuguang Cui. Pointmatch: A consistency training

- framework for weakly supervised semantic segmentation of 3d point clouds. arXiv preprint arXiv:2202.10705, 2022.
- [224] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1912–1920, 2015.
 - [225] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1912–1920, 2015.
 - [226] Zhonghua Wu, Yicheng Wu, Guosheng Lin, Jianfei Cai, and Chen Qian. Dual adaptive transformations for weakly supervised point cloud segmentation. In European Conference on Computer Vision, pages 78–96. Springer, 2022.
 - [227] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3D point cloud understanding. In European Conference on Computer Vision, pages 574–591, 2020.
 - [228] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4606–4615, 2018.
 - [229] Yuwen Xiong, Mengye Ren, Renjie Liao, Kelvin Wong, and Raquel Urtasun. Deformable filter convolution for point cloud reasoning. arXiv preprint arXiv:1907.13079, 2019.
 - [230] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation. In European Conference on Computer Vision, pages 1–19, 2020.
 - [231] Jinfeng Xu, Xianzhi Li, Yuan Tang, Qiao Yu, Yixue Hao, Long Hu, and Min Chen. Casfusionnet: A cascaded network for point cloud semantic scene completion by dense feature fusion. arXiv preprint arXiv:2211.13702, 2022.

- [232] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In International Conference on Machine Learning, pages 2048–2057, 2015.
- [233] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5661–5670, 2020.
- [234] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13706–13715, 2020.
- [235] Kunping Yan, Qingyong Hu, Hanyun Wang, Xiaohong Huang, Li Li, and Song Ji. Continuous mapping convolution for large-scale point clouds semantic segmentation. IEEE Geoscience and Remote Sensing Letters, 2021.
- [236] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5589–5598, 2020.
- [237] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3D object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7652–7660, 2018.
- [238] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3D instance segmentation on point clouds. In Advances in Neural Information Processing Systems, volume 32, pages 6740–6749, 2019.
- [239] Bo Yang, Sen Wang, Andrew Markham, and Niki Trigoni. Robust attentional aggregation of deep feature sets for multi-view 3D reconstruction. International Journal of Computer Vision, 128(1):53–73, 2020.
- [240] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel

- subset sampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3323–3332, 2019.
- [241] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3D recurrent neural networks with context fusion for point cloud semantic segmentation. In European Conference on Computer Vision, pages 403–417, 2018.
- [242] Zhen Ye, Yusheng Xu, Rong Huang, Xiaohua Tong, Xin Li, Xiangfeng Liu, Kuifeng Luan, Ludwig Hoegner, and Uwe Stilla. LASDU: A large-scale aerial LiDAR dataset for semantic labeling in dense urban areas. ISPRS International Journal of Geo-Information, 9(7):450, 2020.
- [243] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3D shape collections. ACM Transactions on Graphics (TOG), 35(6):1–12, 2016.
- [244] Qian Yin, Qingyong Hu, Hao Liu, Feng Zhang, Yingqian Wang, Zaiping Lin, Wei An, and Yulan Guo. Detecting and tracking small and dense moving objects in satellite videos: A benchmark. IEEE Transactions on Geoscience and Remote Sensing, 2021.
- [245] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point completion network. In 2018 International Conference on 3D Vision (3DV), pages 728–737, 2018.
- [246] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. Advances in Neural Information Processing Systems, 34, 2021.
- [247] Wenxiao Zhang and Chunxia Xiao. PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12436–12445, 2019.
- [248] Yachao Zhang, Zonghao Li, Yuan Xie, Yanyun Qu, Cuihua Li, and Tao Mei. Weakly supervised semantic segmentation for large-scale point cloud. In

- Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 3421–3429, 2021.
- [249] Yachao Zhang, Yanyun Qu, Yuan Xie, Zonghao Li, Shanshan Zheng, and Cuihua Li. Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 15520–15528, 2021.
 - [250] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9601–9610, 2020.
 - [251] Yifang Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022.
 - [252] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3D features on any point-cloud. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10252–10263, 2021.
 - [253] Zhenchao Zhang, Markus Gerke, George Vosselman, and Michael Ying Yang. A patch-based method for the evaluation of dense image matching quality. International journal of applied earth observation and geoinformation, 70:25–34, 2018.
 - [254] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1607–1616, 2019.
 - [255] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. PointWeb: Enhancing local neighborhood features for point cloud processing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5565–5573, 2019.

- [256] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 16259–16268, 2021.
- [257] Kang Zhiheng and Li Ning. PyramNet: Point cloud pyramid attention network and graph embedding module for classification and segmentation. arXiv preprint arXiv:1906.03299, 2019.
- [258] Jia-Xing Zhong, Kaichen Zhou, Qingyong Hu, Bing Wang, Niki Trigoni, and Andrew Markham. No pain, big gain: Classify dynamic point cloud sequences with static models by fitting feature-level space-time surfaces. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022.
- [259] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2921–2929, 2016.
- [260] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 633–641, 2017.
- [261] Xiaoyu Zhu, Jeffrey Chen, Xiangrui Zeng, Junwei Liang, Chengqi Li, Sinuo Liu, Sima Behpour, and Min Xu. Weakly Supervised 3D Semantic Segmentation Using Cross-Image Consensus and Inter-Voxel Affinity Relations. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 2814–2824, 2021.
- [262] Xiaoyu Zhu, Jeffrey Chen, Xiangrui Zeng, Junwei Liang, Chengqi Li, Sinuo Liu, Sima Behpour, and Min Xu. Weakly supervised 3D semantic segmentation using cross-image consensus and inter-voxel affinity relations. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2834–2844, 2021.
- [263] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9939–9948, 2021.

- [264] SM Zolanvari, Susana Ruano, Aakanksha Rana, Alan Cummins, Rogerio Eduardo da Silva, Morteza Rahbar, and Aljosa Smolic. DublinCity: Annotated LiDAR point cloud and its applications. In British Machine Vision Conference, 2019.
- [265] Chuhan Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In Proceedings of the IEEE International Conference on Computer Vision, pages 900–909, 2017.