2024-04-23

Self-Supervised Learning Method for Semantic Segmentation of LiDAR Point Clouds

Mutlu Kipirti, Fatma

UNIVERSITY OF CALGARY

Self-Supervised Learning Method for Semantic Segmentation of LiDAR Point Clouds

by

Fatma Mutlu Kipirti

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN GEOMATICS ENGINEERING

CALGARY, ALBERTA

APRIL, 2024

Abstract

Semantic segmentation has shown a significant success for achieving comprehensive scene understanding in real-time perception and urban modeling. Over the recent years, there have been significant advancements in semantic segmentation for LiDAR point clouds, largely the adopting of deep learning techniques. There are the related works of 3D semantic segmentation, including neural network models to process converted voxels, points, and graphs. However, point-based methods are not taken into account local structure feature, resulting in a lack of fine-grained features and limited generalization. Additionally, these models do not take full advantage of the high-level geometric correlations among local neighbors, resulting in low semantic segmentation accuracy. The use of voxel-based methods for balancing precision and computational efficiency is a useful technique. Still, voxel-based representation of point clouds is inefficient and tends to ignore fine details. Little research has investigated using graph-based methods for LiDAR point cloud semantic segmentation. Our work demonstrates the feasibility of using graph representation for highly accurate semantic segmentation in a point cloud. Other problems in the existing methods are high computational and memory requirements. Self-supervised learning on large unlabeled datasets is one way to reduce the number of manual annotations needed. In this thesis, we explore that leverage the combination self-supervised contrastive learning and graph-based method to overcome the semantic segmentation challenges of large-scale point clouds. An experimental qualitative and quantitative analysis of our method shows that the proposed approach can beat previous approaches on S3DIS and SemanticKITTI datasets for the task of LiDAR point cloud semantic segmentation.

Preface

This thesis is original, unpublished, independent work by the author, Fatma MUTLU KIPIRTI.

Acknowledgements

I am grateful for different groups of people who support me through all along the journey of this work.

First of all, I would like to express my deep gratitude to my supervisor, Prof. Ruisheng Wang for valuable advice, good wishes and encouragement.

In addition, many thanks to my fellow lab mates in Spatial Intelligence Lab at the University of Calgary, for establishing a collaborative and efficient environment and always being ready to help.

Most importantly, I would like to thank my family, including my parents, my sisters, and my spouse for all of the love, support, encouragement and prayers they have sent my way along during this process. To my kids, Umut Bugra and Ada, you are my inspiration to achieve this success. I am very grateful for the sacrifices you made for me.

Lastly, I am thankful to The Republic of Türkiye Ministry of National Education for their financial support during master program.

Thank you.

Fatma Mutlu Kipirti

April 2024

Table of Content

List of Tables

List of Figures

List of Abbreviation

2D**---**Two-dimensional

3D**---**Three-dimensional

LiDAR**---**Light Detection and Ranging

CNNs **---** Convolutional Neural Networks

SSL **---** Self-Supervised Learning

DGCNN**---**Dynamic Graph CNN

S3DIS **---** The Stanford Large-Scale 3D Indoor Spaces Dataset

DA **---** Domain Adaptation

MAE **---** Mask Autoencoder

BERT **---** Bidirectional Encoder Representations from Transformers

dVAE **---** Variational Autoencoder

MaskSurf **---** Mask Surfel Prediction

CSC **---** Contrastive Scene Contexts

MPCT **---** Mask point cloud transformer

STRL **---** Spatio-temporal Representation Learning

FPS **---** Farthest Point Sampling

KNN**---**K-Nearest Neighbour

VAE **---** Auto-encoder

RBF **---** Radial Basis Function

VFE **---** Voxel Feature Encoding

3D CNN**---**3D Convolutional Neural Network

MAELi **---** Masked Autoencoder for LiDAR Point Clouds

SPG **---** Super-Point Graph

GACNet**---** Graph Attention Convolution Network

TGCov **---** Geometric Graph Convolution

LAE-Conv **---** Local Attention Edge Convolution

LE-SPG **---** Local Embedding Superpoint Graphs

GIGCN **---** Gated Integration Graph Convolutional Network

EdgeConv **---** Edge Convolution

NT-Xent loss **---** Normalized Temperature-scaled Cross Entropy Loss

IMU**---**Initial Measuring Unit

Adam**---**Adaptive Moment Estimation

IoU**---**Intersection over Union

KNN**---**K-Nearest Neighbour

mIoU**---**Mean Intersection over Union

MLP**---**Multilayer Perception

CHAPTER 1: INTRODUCTION

1.1 Introduction

Several deep learning methods have shown success in point cloud processing, such as object detection, instance segmentation, and semantic segmentation. The availability of large-scale point cloud datasets provides the development of deep learning models for countless applications in computer graphics and vision. With the rapid advancements of 3D LiDAR (Light Detection and Ranging) sensors and acquisition techniques, point clouds have become widely available, prompting research interests in 3D scene understanding. LiDAR sensors capture more precise depth measurements and are more robust against various lighting conditions compared to visual cameras.

The semantic segmentation of 3D point clouds is crucial for enhancing scene understanding in intelligent systems, including robotics [1], [2], [3], [4] autonomous navigation in the context of automatic driving [5], [6], [7], [8], [9], [10], and interaction tasks within real-world environments. The semantic segmentation of a point cloud involves labeling each point with a semantic label, such as person, tree, road, vehicle, ocean, or building. During segmentation, similar points are grouped into homogeneous regions that represent specific structures or objects in a point cloud scene. However, achieving 3D semantic segmentation through the semantic annotation of points presents significant challenges because of that the data structure of the 3D point cloud is irregular, continuous, unstructured, and unordered. To deal with these problems, there are related works of 3D semantic segmentation, including neural network models to process converted voxels, points, and graphs. Although many of these have been successfully tackled using deep learning frameworks, there still exist limitations.

On the other hand, supervised deep learning approaches have been demonstrated to learn semantic information from LiDAR point clouds by assigning a semantic label to each point to represent to what semantic class it belongs. Even though these approaches demonstrate excellent performance in scene understanding, they are often expensive. Therefore, it is of great interest to Self-Supervised Learning (SSL) methods, which can reduce the number of annotated samples.

As a result, the primary aim of this thesis is to design a deep learning framework tailored for extracting 3D information from large scale and irregular point clouds and achieve better results in terms of both accuracy and efficiency compared to the current state-of-the-art 3D deep models. Then, to reduce the dependence on data annotation, we include self-supervised contrastive learning channels for LiDAR point cloud semantic segmentation.

1.2 Motivations

During the last few decades, convolutional neural networks (CNNs) have rapidly developed due to their ability to discriminate features and to generalize [11]. It is commonly understood that these models work best with structured data, such as the 2D pixel arrays [12]. Point clouds present a significant challenge to traditional 2D models due to their irregular and unstructured data formats. The previous attempts at using deep learning for large 3D data attempted to replicate successful CNN architectures used in image segmentation. Specifically, point clouds have the following main properties that make it difficult to learn meaningful features, as visualized in Figure 1:

1. Irregularity: Point cloud data exhibit irregularity, indicating that points are not uniformly sampled across different regions of an object or scene. This results in some regions having densely populated points while others have sparser points, as illustrated in Figure 1a.

Although irregularity can be mitigated to some extent through subsampling techniques, complete elimination is not achievable.

2. Unstructured: Point clouds, unlike structured data, such as images, lack a specific structure. Each point in point clouds is scanned independently, and distance from neighboring points is not always fixed, as seen in Figure 1b.

3. Unordered: A point clouds is an unordered set of vectors, as seen from a data structure point of view, unlike pixel array in images, as displayed in Figure 1c. It means that there will be no change to the actual scene represented by the point cloud when the order of the points is changed.

4. Sparsity: LiDAR point clouds are often sparse, meaning gaps or missing points in the data may exist. Handling sparse data while maintaining accurate segmentation is a challenge.
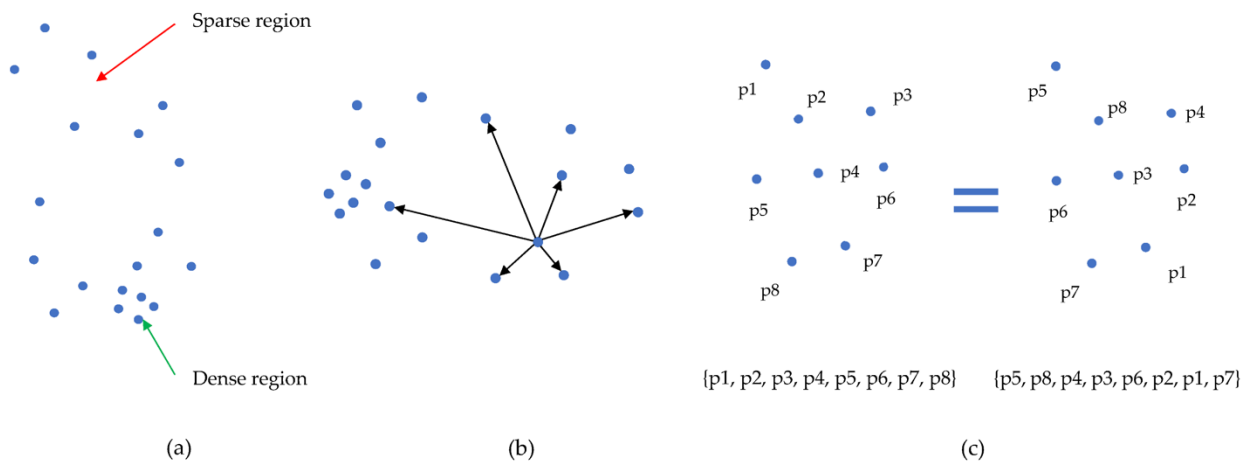
Figure 1: The challenges related to 3D point cloud processing.

To deal with these challenges, several studies based on different representations of point clouds have been applied by using deep learning methods.

View-based 3D models can leverage designed 2D deep architectures and datasets. However, when converting 3D point clouds to 2D space, some geometrically related spatial information can be lost.

Point clouds are divided into regular grids and the 3D point cloud into cubes to describe the LiDAR point clouds in 3D space using voxel-based methods [13], [14]. However, these methods may cause gaps or missing points in the data. Also, computation costs increase with increasing input data size or resolution, limiting the model's performance in dense or large point clouds.

Voxel grids and images are Euclidean-structured data that are suitable for using traditional 2D models to extract distinctive spatial features, including edges and key points [15]. However, it is not possible to accurately maintain the original geospatial information in 3D space during point cloud projection or voxelization. For example, depth information along Z axes is lost when converting point clouds from 3D scenes into (x, y) 2D images. Despite point clouds preserving the original 3D geospatial information, conventional 2D models cannot apply to them due to their unstructured and irregular data. Therefore, Qi et al. proposed the first point cloud-based deep model (PointNet) [16], directly using the point clouds as input. Many deep learning models have been developed based on PointNet to extract the geospatial structure features of a point cloud, including PointNet++ [17] and PointCNN [18]. These approaches contribute to advancing the application of deep learning in tasks related to extracting 3D geospatial information with robust and efficient performances. However, the lack of high-level geometric correlations of local neighbors in these models limits their semantic segmentation accuracy and leads to the complexity of per-point labeling.

In graph-based methods, each graph node represents a point, while the edges represent the relationship between neighbors [19], [20]. Graph-based models have an advantage in leveraging

geometric relationships among points and their neighbors. This advantage allows for extracting spatially local correlation features from the grouped edge relationships.

Deep learning models have made significant progress by developing more robust and efficient deep learning models, data preprocessing techniques, and algorithms specifically tailored for semantic segmentation of large-scale LiDAR point clouds, as explained above. However, there are still various challenging problems, such as high computational and memory requirements to collect point annotations for semantic segmentation of large-scale LiDAR point cloud data, as follows.

1. Labeling and Annotation: Annotating large-scale LiDAR point cloud data with precise labels for semantic segmentation is a time-consuming and labor-intensive task. Developing efficient labeling methods and tools is essential for creating high-quality training datasets.

2. Data Size and Scalability: Large-scale LiDAR point cloud data can be massive, making it computationally expensive and memory-intensive to process and segment. Efficient techniques for handling such data while maintaining high accuracy are essential.

These challenges emerge due to the prevalent use of supervised learning techniques. To deal with these problems, SSL with unlabeled point clouds has drawn much attention in various computer vision tasks.

There are four main requirements for point-based semantic segmentation:

1) Geometric variations should be considered when designing models.

2) Due to the incomplete shape of most 3D objects, the deep models proposed should accurately predict semantic labels with missing information.

3) Exploring local geometric correlations between the input and its neighbors can be challenging. The proposed models should learn such geometric information.

4) The model should leverage large amounts of unlabeled data.

To meet these requirements, the best approach is to convert point clouds to graphs and use a self-supervised contrastive learning method. Thus, this thesis mainly focuses on a graph representation of point cloud data and a self-supervised method for point cloud semantic segmentation when developing deep learning models in indoor and outdoor scenes.

1.3 Contributions

Our work focuses on converting point clouds to graph representations suitable for deep learning without destroying geometric information. Specifically, we connect neighboring points in a point cloud to form an undirected graph. We use the underlying relationship between vertices and their neighborhood to learn meaningful features for segmentation. Graph-based methods have shown appreciable results [21], [22], [23] on point cloud learning algorithms. Graph neural networks process the constructed graphs.

In this thesis, we investigate the possibility of employing Dynamic Graph CNN (DGCNN) [21] with self-attention for self-supervised semantic segmentation from LiDAR point clouds, taking into account global features and local features to enhance the feature learning process.

In summary, we make the following key contributions to this work.

1. We propose an improved approach using graph representation of LiDAR point cloud.

2. To reduce the dependence on data annotation, we develop self-supervised contrastive learning channels for LiDAR point cloud semantic segmentation. Further improving the accuracy of our feature extraction network, we apply dual scales of point-level and graph-level contrastive learning strategies.

3. We utilize DGCNN to get the local shape of the point cloud as the feature extractor for the entire network. We add self-attention after EdgeConv to leverage global features to perform segmentation on LiDAR point clouds.

1.4 Structure of the Thesis

The purpose of this thesis is to examine the challenges and opportunities associated with deep learning and semantic segmentation of 3D point clouds in indoor and outdoor environments. This thesis is arranged as follows:

Chapter 2 provides an overview of both indoor and outdoor LiDAR datasets used for training 3D deep models for segmentation and detection tasks. In addition to, the knowledge of SSL and existing deep learning studies related to methods of point cloud segmentation are provided.

Chapter 3 introduces methodology by briefly reviewing DGCNN, contrastive learning, and loss function to understand the proposed pipeline.

Chapter 4 details a proposed graph architecture for per-point semantic segmentation in indoor and outdoor scenes to get the geometric attributes among local points to improve the segmentation results. In addition to evaluating the algorithm's accuracy and efficiency, several evaluation metrics for semantic segmentation are presented in order to compare it with existing state-of-the-art methods.

Chapter 5 introduces semantic segmentation results and discusses how to boost 3D semantic segmentation accuracy in indoor and outdoor scenes.

Chapter 6 summarizes this research with a summary of contributions and details of future works.

CHAPTER 2: BACKGROUND AND RELATED WORKS

2.1 3D LiDAR Datasets

Autonomous vehicles increasingly rely on LiDAR sensors. The LiDAR sensor uses lasers to scan

its surroundings and then produces a point cloud. Datasets of 3D LiDAR are divided into three

groups based on the data acquisition method and the main application, as listed in Table 1. Figure

3 also shows five datasets that are commonly used for point cloud semantic segmentation.

Table 1: 3D LiDAR Datasets with Comparison *[24]*.

Sensor	Data Type	Anno.	Dataset	Frames	Points	Classes	Year	Organization
3D LiDAR	Static	Point	Oakland	17	1.6M	44	2009	CMU
			Paris-rue-Madame	2	20M	17	2014	MINES ParisTech
			TerraMobilita/Iqmulus	10	12M	15	2015	Univ. of Paris-Est
			S3DIS	5	215M	12	2016	Stanford Univ.
			TUM City Campus4	631	41M	8	2016	TUM
			Semantic3D	30	4009M	8	2017	ETH Zurich
			Paris-lille-3D	3	143M	50	2018	MINES ParisTech
	Sequential	Point	Sydney Urban	631	/	26	2013	ACFR
			SemanticKITTI	43552	4549M	28	2019	Univ. of Bonn
			SemanticPOSS	2988	216M	14	2020	Peking Univ.
			A2D2	41277	1238M	38	2020	Audi
			PandaSet	16000	1844M	37	2020	Hesai & Scale
			muScenes-lidarseg	40000	1400M	32	2020	nuTonomy
			KITTI-360	100K	18B	19	2020	Univ. of Tübingen
		3D-Box	KITT	14999	1799M	8	2012	KIT
			H3D	27K	/	8	2019	HRI
			nuScenes	40K	2780M	23	2019	nuTonomy
			Lyft L5	46K	9936M	9	2019	Lyft Inc.
			Arcovers	22K	2354M	15	2019	Argo AI
			Waymo	230K	40710M	4	2020	Waymo LLC
			A*3D	39K	5093M	7	2020	I2R
			DENSE	13.5K	/	4	2020	Mercedes-Benz
	Synthetic	Point	GTA-V	/	/	/	2018	UC, Berkeley
			SynthCity	75000	367.9M	9	2019	UCL

1) Static Datasets: Data collected from static viewpoints by using terrestrial laser scanners or mobile laser scanning (MLS) systems that mainly capture static scene objects for applications including street view, 3D modeling, and virtual realities. The primary application scenarios for these datasets involve urban planning, robotics, and augmented reality. Semantic3D [25] is the largest and most popular outdoor dataset in the static dataset, while the Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) [26] is a large dataset of indoor scenes.

2) Sequential Datasets: Data is collected as sequences of frames   by using vehicular platforms for autonomous driving tasks. Autonomous driving systems are leveraged to capture the sequences of LiDAR frames with a moving viewpoint on the street. These datasets usually contain more frames and sparse points compared to static datasets. Sequential datasets with both point-wise and instance labels have significantly contributed to advancing research in the field of 3D semantic segmentation [27] and panoptic segmentation [28]. SemanticKITTI [29] is the largest and most popular sequential dataset, while SemanticPOSS [30] is a new dataset describing a dynamic urban scene with rich cars, people, and riders.

3) Synthetic Datasets: Synthetic datasets are collected in a virtual world by simulating any data acquisition systems. Generating real datasets is prohibitively expensive, leading to the creation of synthetic datasets through computer simulation. These synthetic datasets can be large-scale at a more affordable cost. However, the challenge with using synthetic datasets arises from the significant gap between synthetic and real scenes. Synthetic scenes are realistic, but they lack accuracy in detail. Data are collected in a virtual world by simulating any of the above data acquisition systems. The generation of real datasets is extremely expensive due to the labor intensiveness of data annotation. Synthetic datasets are built through computer simulation, which can be large scale and have fine but cheap annotations. The large gap between synthetic and real

scenes causes the problem of using such datasets. Although synthetic scenes can generally be very realistic, they lack the necessary details of real objects. Figure 2 shows five datasets commonly used for point cloud semantic segmentation.

(a) ShapeNet   (b) S3DIS   (c) ScanNet

(d) Semantic3D   (e) SemanticKITTI
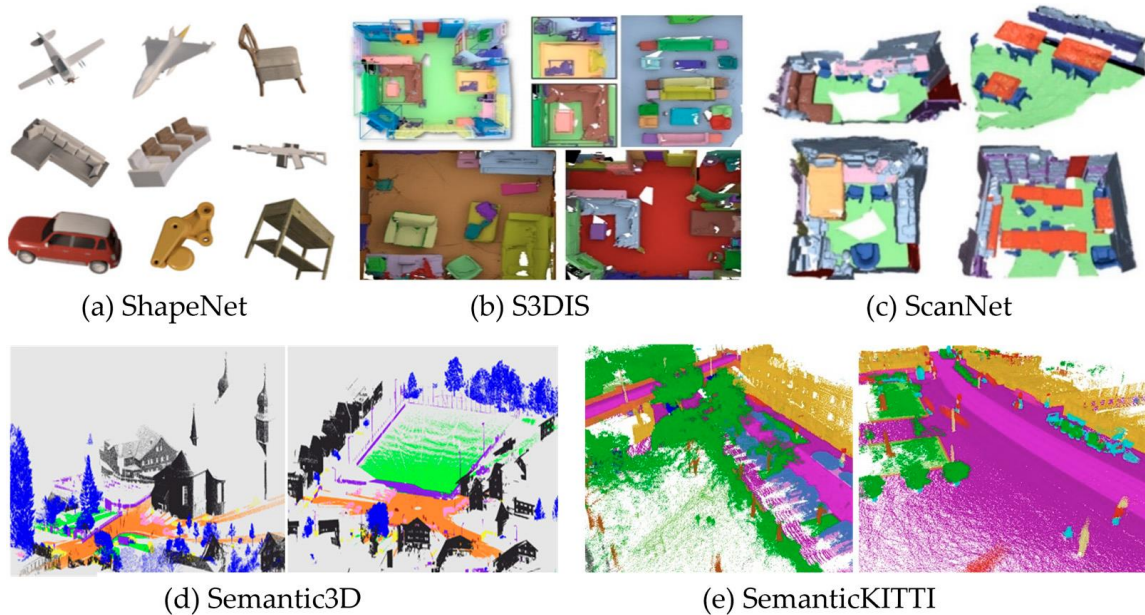
Figure 2: Examples of ShapeNet (a), S3DIS (b), ScanNet (c), Semantic3D (d), and SemanticKITTI (e).

2.2 Self-Supervised Learning

Self-supervised learning is a type of unsupervised learning in which no supervision is given at all, where the labels are generated from the data itself [31], [32], [33]. Not only does it solve the problem of the error-prone and expensive labeling process, but it also relieves the domain adaptation (DA) issues [34] with improved model generalization ability. The primary objective of SSL is to pre-train an encoder on an unlabeled, large-scale point cloud dataset and then transfer the well-trained network to other datasets for various downstream tasks. A comprehensive SSL framework typically consists of the following essential modules.

• Data augmentation: The raw input is augmented by adding noise, translation, rotation, flipping, and flipping horizontally [35]. The large gap between synthetic and real scenes causes the problem of using such datasets.

• Encoder: The encoder for point clouds typically uses a neural network architecture that captures the hierarchical representation of the input point cloud data.

• Pretext task: In self-supervised learning for point clouds, pretext tasks are designed to extract the hidden self-supervision signal via the interactions between the encoder and data.

• Knowledge transfer: The encoder is transferred with the knowledge gained during pretext task pretraining to downstream tasks.

• Downstream task: Evaluation metrics and experimentation with different downstream tasks such as object classification, semantic segmentation, and object detection are often necessary to assess the effectiveness of the SSL framework.

Existing self-supervised learning methods on point clouds can be classified as reconstruction-based, contrast-based, alignment-based, and motion-based methods, as shown in Figure 3.
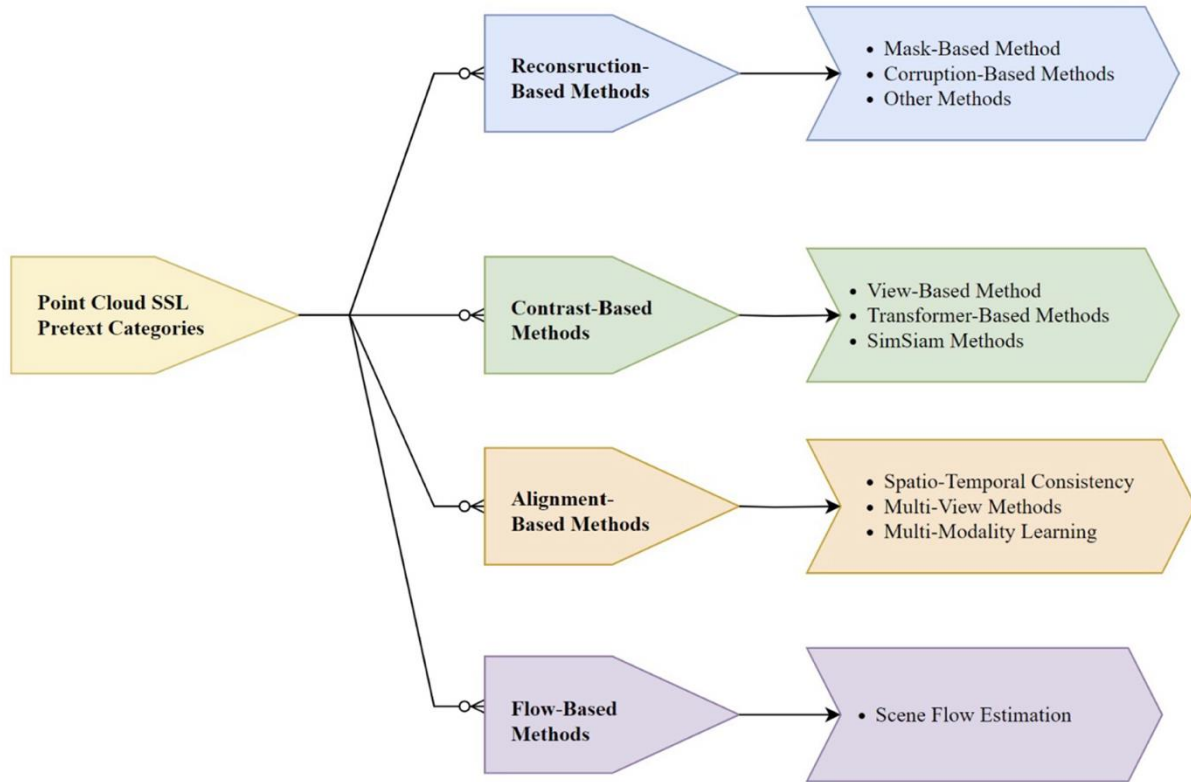
Figure 3: Taxonomy of SSL for point cloud data based on pretext tasks.

2.2.1 *Reconstruction-based Methods*

A reconstruction-based method learns point cloud representations by reconstructing corrupted point clouds and restoring them as much as possible. Global features, as well as the mappings between local and global areas, are learned during the reconstruction process. The fundamental concept behind reconstruction involves masking a portion or portions of the point cloud and subsequently recovering the missing parts through an encoder-decoder architecture. Reconstruction-based self-supervised learning methods can be classified into two major subgroups: mask-based methods and corruption-based methods. A summary of the methods under these two sub-categories and other methods is shown in Table 2.

Table 2: Summary of reconstruction-based point cloud SSL methods

Year	Methods	Sub-categories	Contributions
2021	Point-BERT [38]	Mask-based Methods	Reconstructing missing point tokens with BERT-style transformer.
2022	Point-MAE [39]	Mask-based Methods	Shifting masked tokens to decoder to avoid early leakage.
2022	MaskSurf [38]	Mask-based Methods	Estimating surfel position and per-surfel orientation simultaneously.
2022	Voxel-MAE [41]	Mask-based Methods	Performing additional binary voxel classification for complicated semantics awareness.
2022	CP-Net [42]	Corruption-based Methods	Disentangling point clouds into contour and content ingredients.
2021	Self-correction [41]	Corruption-based Methods	Recovering shape-disorganized point regions.
2022	Point-DAE [35]	Corruption-based Methods	Exploring three corruption families such as density/masking, noise, and affine transformation.
2022	SeRP [42]	Corruption-based Methods	Using perturbed point clouds as input.
2019	3D jigsaw [43]	Other Methods	Rearranging randomly disorganized point clouds.
2021	DefRec [44]	Other Methods	Performing deformation on 2D grids to fit arbitrary 3D object surface.
2022	UAE [45]	Other Methods	Gaining both advanced semantic information and basic geometric structure.

Similar to Mask Autoencoder (MAE)[36], the encoder plays a crucial role in capturing the local geometric structure and regional relations during the restoration process. Point-BERT [38] is derived from BERT (Bidirectional Encoder Representations from Transformers)[37]. Point-BERT focuses on point clouds and incorporates a point-specific tokenizer based on discrete Variational Autoencoder (dVAE), which enables the model to map patches in point clouds to discrete tokens, facilitating the capture of meaningful local geometric patterns. However, Point-BERT has limitations, including pretraining dependency, over-reliance on contrastive learning, dependency

on data augmentation, task specificity, and computational requirements. To deal with some of these issues, Point-MAE[39], as proposed by Pang et al. in 2022, tackles the challenge of processing point clouds with a high ratio (60%-80%) of randomly masked points. Point-MAE includes a standard transformer as its backbone with an asymmetric encoder-decoder architecture to process random masking points. By shifting the mask tokens from the encoder to a lightweight decoder, Point-MAE saves computational resources and prevents early leakage of location information. Zhang et al. propose Mask Surfel Prediction (MaskSurf) [40] to use surfels to represent the local geometry of point patches, allowing the model to acquire more information regarding the local structure of point clouds in the self-supervised learning process. Voxel-MAE [41] combines the advantages of voxelization and mask-based pretraining. In addition to reconstructing the occupancy value for masked voxels, a supplementary binary voxel classification task distinguishes whether a voxel contains points clouds enhances the model's abilities to learn complex semantics.

To deal with leakage of location information and uneven information density, the SeRP [42] algorithm uses encoder-decoder architecture to reconstruct the original point cloud based on a perturbation technique. 3D jigsaw [43] was proposed as a 3D version of the jigsaw pretext to restore the original position of each patch from disorderly distribution.

2.2.2 *Contrast-based Methods*

Contrastive learning is a popular self-supervised learning method that includes view-based methods, transformer-based methods, and SimSiam-based methods. Instead of depending on the specifics of individual samples, contrastive learning focuses on overall semantic learning through discriminative pretext tasks that capture context similarity and difference of point clouds. In addition to making contrastive learning models easier to optimize, this characteristic expands the

views of input point clouds by various data augmentation techniques. This section discusses the

contributions and limitations of contrast-based methods. A brief summary of these methods is

shown in Table 3.

Table 3: Summary of contrast-based point cloud SSL methods.

Year	Methods	Sub-categories	Contributions
2020	PointContrast [51]	View-based Methods	Obtaining dense features at point-level on complex scenes by point contrast
2020	P4Contrast [52]	View-based Methods	Utilizing synergies between two modalities for better feature extraction
2021	Contrastive Scene Contexts [53]	View-based Methods	Introducing ShapeContext local descriptor and achieving data-efficiency
2021	SegContrast [54]	View-based Methods	Structural information and a more descriptive feature representation
2021	DepthConstrast [52]	View-based Methods	Applying Instance Discrimination on depth maps
2022	AFSRL [55]	View-based Methods	Imposing data-level augmentation and feature enhancement simultaneously
2022	ContrastMPCT [56]	Transformer-based Methods	Unnecessary to pre-train a "tokenizer" and makes it easier to train.
2022	POS-BERT [57]	Transformer-based Methods	Maximizing the class token consistency among point cloud pairs.
2022	Distillation with Contrast [58]	Transformer-based Methods	Utilizing knowledge distillation and contrastive learning.
2023	RECON [58]	Transformer-based Methods	Unifying masked generative modeling and contrastive modeling.
2021	SimSiam [59]	SimSiam-based Methods	Using only positive sample pairs and shared weights.
2022	ConClu [60]	SimSiam-based Methods	Reducing the reliance on negative samples in contrastive learning
2022	4dcontrast [63]	SimSiam-based Methods	Incorporating 4D sequence information and constraints into 3D representation learning

PointContrast[52], a popular contrastive-learning method, allows performing point-level

comparisons in two transformed point clouds with different views to capture dense information at

the point level. However, there are still several areas for improvement. A major limitation of this method is that it ignores spatial contextual information such as orientation, distance, and relative position, which play a significant role in many understanding tasks. As a second limitation, PointContrast can only be scaled up to 1,024 points for pretraining, so providing more points will not result in better performance. Moreover, PointContrast requires resource-intensive inputs, such as the absolute position of the camera, which is difficult to obtain. Zhang et al. introduced DepthContrast[52] as a solution to alleviate the resource-intensive challenges associated with PointContrast, by employing only a single-view depth map. To address the data-efficiency problem in 3D scene understanding, Hou et al. presented Contrastive Scene Contexts (CSC) [54] to fuse spatial information into pretraining objects by introducing ShapeContext local descriptor [51] partitioning and performing contrastive learning in each region. The method outperforms when compared to PointContrast in semantic segmentation. To leverage the invariances of 3D features, Li and Heizmann [53] proposed a contrastive learning framework with jointly pre-training 3D encoders and depth graph encoders, unifying modality-invariance between RGB and depth images, and format-invariance between point clouds and voxels. With aiming to learn the intrinsic features of point clouds at various sampling patterns and densities, Chen et al. [54]proposed a multilevel self-supervised learning method for geometric sampling invariant representation.

SegContrast [54] is an approach developed by Nunes et al. that employs self-supervised segment discrimination to address the challenge of learning 3D point cloud feature representations. A two-stage process is used in this method, with the first stage being self-supervised and free of labeled data. The model takes the feature representation obtained from the self-supervised learning in the first stage and refines it through fine-tuning in the second stage. An overview of SegContrast's

framework is shown in Figure 4. SegContrast contributes a contrastive representation learning method for 3D LiDAR point clouds that can learn structural information and a more descriptive feature representation. By using contrastive learning, it can discriminate between similar and dissimilar structures based on the LiDAR data segments. The evaluations show better results in describing fine-grained structures.
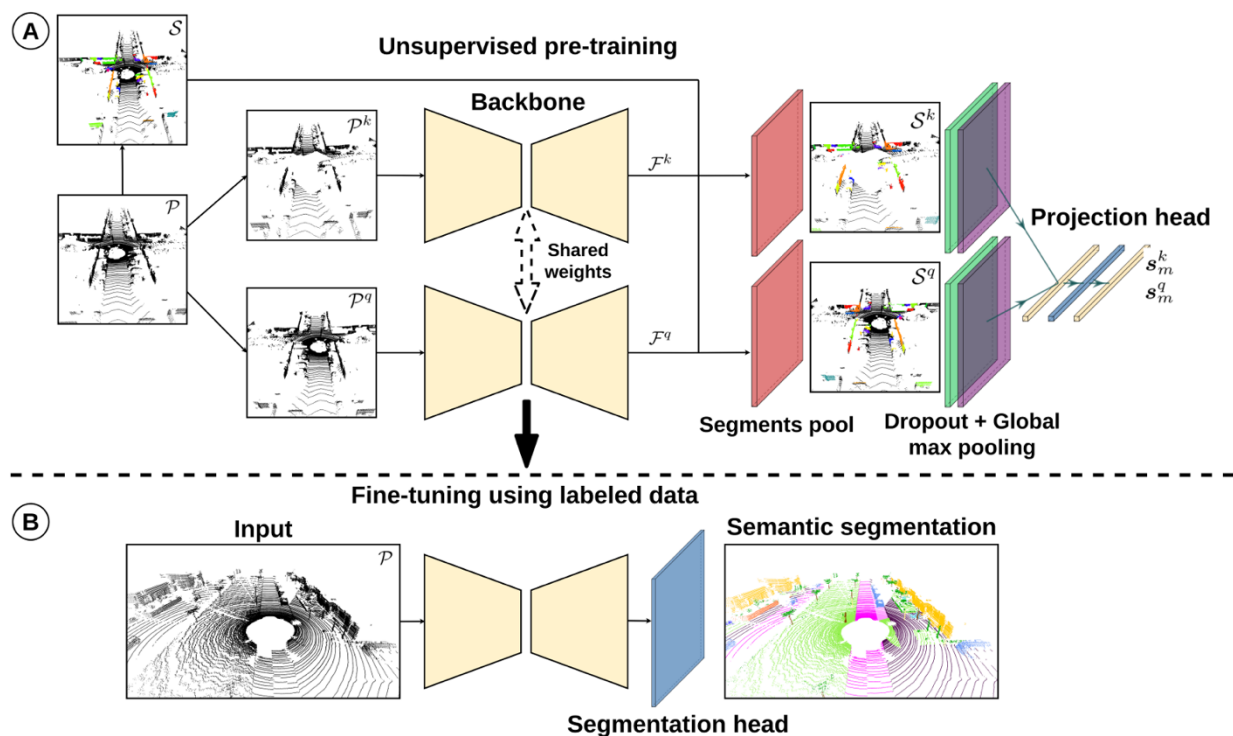
Figure 4: A popular example of contrastive learning methods, SegContrast *[54]*.

Transformer-based models in 3D point cloud pre-training have attracted increasing attention in recent years because of applying on unordered points without the need for any explicit positional encoding. These models efficiently capture long-range dependencies between elements by utilizing self-attention mechanisms. In Mask point cloud transformer (MPCT), input points are randomly masked and then recovered using a Transformer. ContrastMPCT [56] is proposed to optimize the reconstruction performance of MPCT, computing the contrast loss between the point

cloud reconstructed by MPCT and the original point cloud. In contrast to Point-BERT, the contrastive learning design of ContrastMPCT makes it unnecessary to pre-train a "tokenizer" and makes it easier to train.

SimSiam-based contrastive learning approaches [59] only require positive sample pairs during training and utilize shared weights to separately encode different augmented versions of the same input. With positive sample pairs and shared weights, SimSiam-based methods reduce training data requirements and improve overall performance.

2.2.3. Alignment-based Methods

Based on the property of point cloud representation generally being invariant to transformations, alignment-based methods, including spatiotemporal consistency, multi-view methods, and multi-modality learning, have been proposed to learn the implicit embedding of point clouds. Table 4 provides a brief summary of the methods in this category.

Table 4: Summary of alignment-based point cloud SSL methods.

Year	Methods	Sub-categories	Contributions
2021	STRL [61]	Spatiotemporal consistency	Dual-branch network to predict representation of another temporally correlated input
2022	Future prediction [62]	Spatiotemporal consistency	Forecasting future point cloud scenes with lightweight model
2020	Info3D [63]	Multi-view methods	Maximizing mutual information between objects and their transformations.
2021	Cross-view [64]	Multi-view methods	Jointly learning both 3D point cloud and 2D image embedding concurrently.
2022	I2P-MAE [65]	Multi-view methods	Combining 2D guided masking strategy and the 2D-semantic reconstruction strategy for the 2D feature map and 2D visual features
2022	Multi-view rendering [66]	Multi-view methods	Encouraging 2D-3D global feature distributions to be similar.
2022	PointCLIP [67]	Multi-modality learning	Fusing few-shot 3D knowledge into 2D models.

2022	PointCLIP V2 [68]	Multi-modality learning	A 4-step projection module and an optimized prompt selection strategy.
2023	CLIP2 [69]	Multi-modality learning	Constructing well-aligned instance-based text-image-point proxies from complex scenarios
2023	ULIP [70]	Multi-modality learning	Aligning a 3D embedding to pre-aligned image-text feature space to obtain unified representation

A spatial-temporal approach emphasizes long-range spatial and temporal invariance for capturing dynamic sequence intrinsic characteristics. Huang et al. introduced the spatio-temporal representation learning (STRL) [61] framework, augmented by random spatial transformation to extract spatial and temporal representations from 3D shapes. By considering training and inference times, Mersch et al. [62] present an auto-encoder architecture that integrates spatial and temporal information using 3D convolutions. Firstly, past point clouds are projected into 2D range images and then concatenated as a spatial-temporal tensor. In this approach, skip connections and horizontal circular padding are employed during convolutions to capture spatial and temporal scene information simultaneously.

Multi-view approaches enhance the robustness and generalization of learned representations by integrating information from multiple viewpoints. For example, the Info3D [63] method proposed by Sanghi in 2020 focuses on acquiring rotation-insensitive representations by maximizing mutual information between 3D objects and their local chunks for patch-level consistency. Jing et al. [64] introduced a multi-view SSL method leveraging cross-modality and cross-view correspondences to learn features from 2D images and 3D point clouds jointly. Zhang et al. proposed I2P-MAE [65] to learn 3D representations from a 2D pre-trained model with two image-to-point strategies: the 2D guided masking strategy and the 2D-semantic reconstruction strategy for the 2D feature map and 2D visual features. Tran et al. [66] presented a dual-branch model that jointly employs local pixel-point-level correspondence loss and global image-point cloud-level loss.

The multi-modal approaches aim to take advantage of the correlations between different modalities, such as images, texts, and point clouds. They also leverage complementary information from multiple sources, robustness to missing or noisy data in any one modality, and improved generalization to new environments. PointCLIP [67], inspired by CLIP [71], which learns transferable visual features with natural language supervision, is a notable example of a multi-modal approach. This method effectively combines few-shot 3D knowledge into 2D models, showcasing its ability to fuse information across different modalities for enhanced performance and understanding. PointCLIP faces challenges related to sparse visual projection and textual prompt complexity. To deal with these limitations, PointCLIP V2 [68], is proposed by Zhu et al., incorporates a 4-step projection module and an optimized prompt selection strategy. ULIP [70] is a method that learns unified representations of three modalities (2D images, text, and 3D point clouds). ULIP employs a two-step approach due to the lack of accessible triplet data. In the initial step, the model undergoes pretraining using a large dataset consisting of image-text pairs. This pretraining phase is crucial for ULIP to capture meaningful and transferable visual as well as textual features, establishing a common vision-language feature space. To contribute to cross-modal downstream tasks in the second step, the model aligns a small set of automatically synthesized point cloud triplets into the pre-aligned visual-language feature space.

2.2.4. *Flow-based Methods*

Flow-based methods aim to dynamically extract the intrinsic motion characteristics from spatial variations of the relative motion of each 3D point in a temporal sequence of point clouds. Scene flow estimation is an important topic in autonomous driving. A brief summary on the methods under this category is shown in Table 5.

Table 5: Summary of flow-based point cloud SSL methods.

Year	Methods	Sub-categories	Contributions
2019	FlowNet3d [72]	Scene flow estimation	Learning deep hierarchical features of point clouds and flow embeddings that represent point motions
2020	Just go with the flow [73]	Scene flow estimation	Optimizing two SSL losses based on nearest neighbors and cycle consistency
2020	Pointpwc-net [74]	Scene flow estimation	Discretizing cost volume onto 3D point clouds in a coarse-to-fine fashion
2021	Self-Point- Flow [75]	Scene flow estimation	Converting pseudo label matching problem as optimal transport task
2021	Slim [76]	Scene flow estimation	Classifying points as 'moving' and 'stationary' based on self-supervised training.
2021	Flowstep3d [77]	Scene flow estimation	The first recurrent architecture for non-rigid scene flow.
2021	3D-OGFlow[78]	Scene flow estimation	Merging two networks across all layers to conduct flow estimation and learn the occlusions simultaneously.
2022	RigidFlow [79]	Scene flow estimation	Decomposing the point cloud into supervoxels and predicting the rigid flow for each supervoxel.

The FlowNet3D [72] network is one of the earliest deep neural networks that estimate motion or flow between different 3D point clouds. Mittal et al. developed a self-supervised training approach for scene flow estimation by optimizing two loss components based on the nearest neighbors and cycle consistency. Nearest neighbor loss measures the mean Euclidean distance between each estimated point and its nearest neighbor, while cycle consistency loss is required to prevent network degeneration and enhance the robustness of the model. However, FlowNet3D has limitations, including a degeneration solution and overlooking other discriminative measures such as colors and surface normal. To address these issues, Self-Point- Flow [75] employs more than 3D point coordinates, surface normal, and color in one-to-one matching. The GC network is trained to segment multiple objects simultaneously in a single forward pass using learned dynamic motion patterns as supervision signals, incorporating object shape invariance and rigid consistency into the loss function to ensure high-quality segmentation.

2.3 Methods of 3D Point Clouds Semantic Segmentation

Many deep learning methods for 3D point clouds semantic segmentation have been proposed in the literature, offering diverse approaches to address the challenges of 3D semantic segmentation. These methods can be categorized into three main groups based on the representation of the data: point-based methods, voxel-based methods, and graph-based methods, as shown in Figure 5.

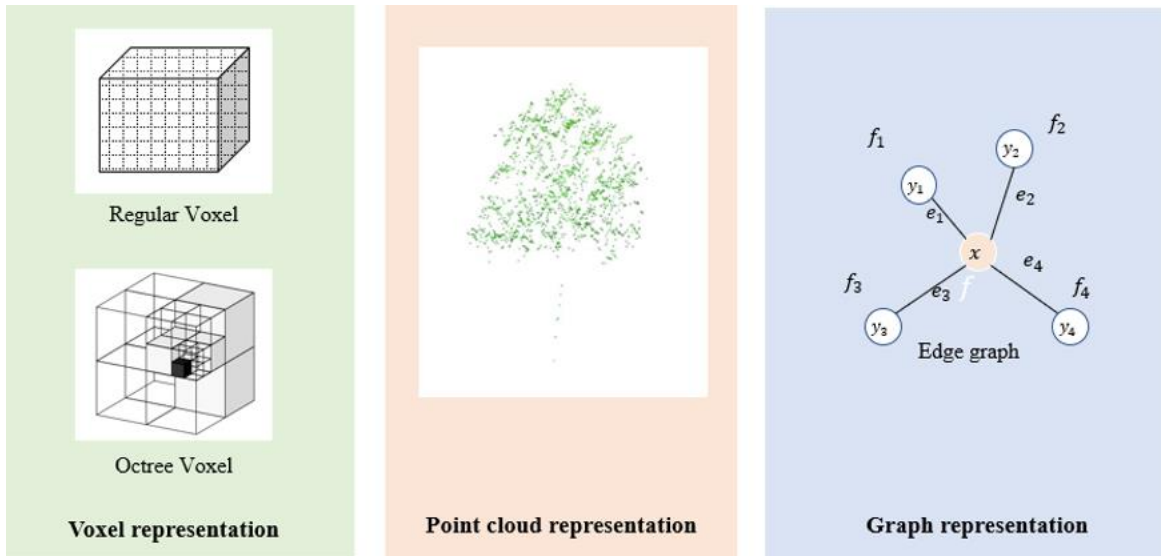Figure 5: Point cloud representations.

2.3.1 *Point-Based Methods*

Point-based methods, in the context of 3D deep learning, capture representations at the point-wise level. Point- based semantic segmentation methods are summarized in Table 6.

Table 6: Summary of point-based semantic segmentation methods with deep learning.

Year	Methods	Sub-categories	Contributions
2017a	PointNet [16]	Point-based Method	Pioneering processing points directly.
2017b	PointNet++ [17]	Point-based Method	Proposing hierarchical learning framework.
2019	KPConv [80]	Point-based Method	Novel point convolution.
2018	Point CNN [18]	Point-based Method	Coarsens the input points with farthest point sampling and an χ-transformation from local points by MLP
2019	PointConv [81]	Point-based Method	Novel point convolution considering point density.
2020	RandLA-Net [27]	Point-based Method	LFAM with large receptive field and keeping geometric details.
2021	Point Transformer [82]	Point-based Method	Combining MLP-based relative position encoding and vector attention.
2022	Point Transformer v2 [83]	Point-based Method	Combining novel position encoding and grid Pooling.
2022	Repsurf [84]	Point-based Method	Local triangular orientation + local umbrella orientation.
2022	PointNeXt [85]	Point-based Method	An inverted residual bottleneck design and employs separable MLPs.

PointNet [16] is a pioneering work that exploits points-wise features. By leveraging the point permutation invariance, it adopts a symmetric function such as max-pooling to collect these features into a global feature representation.

However, adapting only symmetric functions like max-pooling in PointNet fails to capture local features, thereby limiting its ability to recognize fine-grained patterns and generalize to complex scenes. Therefore, Qi et al. developed PointNet++ [17] that adopts a multi-stage hierarchical learning approach that involves sampling points using farthest point sampling (FPS), grouping local regions using k nearest neighbor (KNN), and utilizing a simplified PointNet to capture features at various scales. Researchers such as Pang et al.[39], Yu et al.[38], and Zhang et al.[40] explored the effectiveness of combining FPS and KNN in their respective research contexts.

PointNeXt [85] is introduced by Qian et al. in 2022, takes a different approach by revisiting the classical PointNet++ architecture. The focus of this is on conducting a systematic study of model training and scaling strategies to improve the overall performance of PointNet++.

2.3.2 *Voxel-Based Methods*

Voxel-based methods convert 3D LiDAR data into voxels for structured data representation to predict each voxel with one semantic label. These methods are summarized in Table 7.

Table 7: Summary of voxel-based semantic segmentation methods with deep learning

Year	Methods	Sub-categories	Contributions
2016	3D FCN [87]	Voxel-based Method	Efficiently handling large data.
2017	SEGCloud [88]	Voxel-based Method	Combining 3D FCNN with fine-grained predictions.
2017	3D CNN-DQN-RNN [89]	Voxel-based Method	Integrating three vision tasks into one frame.
2018	FCPN [87]	Voxel-based Method	First fully convolutional network on raw point sets.
2018	Scancomplete [90]	Voxel-based Method	Combing scene completion and semantic labeling.
2019	Vv-net [91]	Voxel-based Method	Using a variational auto-encoder (VAE) taking radial basis function (RBF).
2019	VoxelNet [92]	Voxel-based Method	Voxel partitioning and VFE layers.
2022	Voxel-MAE [41]	Voxel-based Method	Adopted a range-aware random masking strategy and designed a binary voxel classification task.
2022	Voxel- MAE [41]	Voxel-based Method	A Transformer-based 3D object detection backbone.
2022	Maeli [93]	Voxel-based Method	Distinguishing between empty and non-empty voxels with a novel masking strategy.
2022	Gd- MAE [94]	Voxel-based Method	Using a generative decoder.

3D Convolutional Neural Network (3D CNN) is a common architecture used to process uniform voxels for point cloud semantic segmentation [95], [96]. However, voxel-based methods based on 3D CNN have these limitations, including finding a proper voxel size that balances precision and computational efficiency. Therefore, several proposed methods such as 3D FCN [87], SEG-Cloud [93], 3DCNN-DQN-RNN [94], FCPN [95], Scancomplete [96], and Vv-net [97] have a focus on addressing these limitations by maintaining acceptable accuracy.

For example, VoxelNet [98] is a generic point-specific network that utilizes voxel partitioning and Voxel Feature Encoding (VFE) layers to capture meaningful features within point clouds. Innovation with VFE Layers plays a crucial role in encoding interactions between points within a

voxel and capturing descriptive appearance information. Despite its innovative features, VoxelNet faces challenges such as the expensive computation of voxel construction and constraining the model from capturing high-resolution or fine-grained representations.

Also, to address the problems associated with processing large-scale point clouds directly, voxel-based SSL methods have emerged as effective solutions. There have been two VoxelMAE methods proposed to improve 3D point cloud perception [41], [46]. The first method is adopted by Min et al. [41] using a range-aware random masking strategy and a binary voxel classification task. Another study [46] trains a Transformer-based 3D backbone to recover obscured voxels and distinguish between free and occluded voxels. Masked Autoencoder for LiDAR point clouds (MAELi), introduced by Krispel et al. [100], detects empty voxels and employs a new masking strategy targeting LiDAR's inherent spherical projection. However, MAE [36] approach still has challenges associated with handling irregular shapes in the context of large-scale 3D point cloud exploration.

2.3.3 *Graph-Based Methods*

Graph-based methods construct a graph from 3D LiDAR data. A graph is a data structure comprising nodes and edges, where nodes represent objects, and edges represent relationships between these objects. Two essential operations in graph-based methods for processing 3D LiDAR data are graph construction and graph convolution. Using graph networks to model local point clouds leverages the connectivity between points, emphasizing the importance of neighbors in capturing the geometry properties of individual points within the local context. This approach contributes to a more comprehensive representation and understanding of the spatial characteristics of the point cloud.

Super-point graph (SPG) [97] builds super graphs on global super-points before applying neural networks to learn per semantics. Furthermore, this network adopts PointNet to encode vertex features and graph convolutions to extract contextual information.

DGCNN [21], proposed by Wang et al., is built upon the PointNet architecture. Instead of directly learning representations for points, DGCNN focuses on capturing interactions between points and their edges in both Euclidean and semantic space. The utilization of DGCNN as a foundational architecture in SSL models highlights its effectiveness in understanding and processing point cloud data [98], [43], [99].

An end-to-end graph attention convolution network (GACNet) [100] captures the structured features of point clouds for fine-grained segmentation. GraphCNNs [21], [100], [101], [102], have been used in several studies to handle unordered point sets with varying neighborhood sizes. Standard graph convolution employs shared weight functions for each pair of points to extract corresponding edge features.

SPH3D-GCN [103] proposes a separable spherical convolutional kernel for graph neural networks, which consists of the spherical convolution learning depth-wise features and point-wise convolution learning point-wise features.

Geometric Graph Convolution (TGCov) [104] extends the capabilities of PointNet++ by introducing a novel graph convolutional operation that integrates local point-wise features with local geometric connection features expressed by Gaussian weighted Taylor kernels.

In the work by Feng et al., the authors focused on constructing a local graph on neighborhood points searched along multidirections and exploring local features using a local attention edge convolution (LAE-Conv) [105].

In their paper, Geng et al. proposed a structural representation algorithm for local embedding superpoint graphs (LE-SPG) [106]. They developed a gated integration graph convolutional network (GIGCN) [107] for feature learning and semantic segmentation graphs. Graph-based point cloud semantic segmentation methods are summarized in Table 8.

Table 8: Summary of graph-based semantic segmentation methods with deep learning.

Year	Methods	Sub-categories	Contributions
2017	GraphSAGE [107]	Graph-based Method	Inductive framework that leverages node feature information.
2018	SPG [97]	Graph-based Method	Superpoint graph + parsing large-scale scene.
2018	LS-GCN [108]	Graph-based Method	Local spectral graph + Novel graph convolution.
2019	DeepGCN [109]	Graph-based Method	Adapting residual connections between layers.
2019	DGCNN [21]	Graph-based Method	Novel graph convolution + updating graph.
2019	HDGCN [110]	Graph-based Method	Depthwise graph Convolution + Pointwise Convolution.
2019	TGNet [111]	Graph-based Method	Novel graph Convolution + multi-scale features exploration.
2019	GACNet [106]	Graph-based Method	Combining novel position encoding and grid Pooling.
2019	LAE-Conv [112]	Graph-based Method	A local graph based on the neighborhood points searched in multi-directions.
2020	Orientation Estimation [98]	Graph-based Method	Considering the auxiliary task of predicting rotations.
2020	SPH3D-GCN [110]	Graph-based Method	Novel graph convolution + pooling + uppooling.
2022	Crosspoint [105]	Graph-based Method	A cross-modal contrastive learning approach.
2023	AF-GCN [112]	Graph-based Method	Combining graph convolution and self-attention mechanisms.
2023	3DGraphSeg [113]	Graph-based Method	Proposing a local embedding super-point graph to alleviate gradient vanishing or exploding.

CHAPTER 3: METHODOLOGY

It is possible to reflect the geometry of a local point cloud rather than working on individual points like PointNet [16]. A graph could express such a local relationship. For example, Wang et al. proposed DGCNN [21] for encoding edge features between vertices, which has demonstrated effectiveness in 3D semantic segmentation. Therefore, using DGCNN as a backbone is a reasonable and appropriate choice for our specific study.

Self-supervised representation learning method could reduce the amount of required labeled data by learning descriptive representations from unlabeled data. Therefore, in this thesis, we leverage unlabeled data with self-supervised learning in 3D semantic segmentation. Our method is based on contrastive learning that is one of the self-supervised learning methods.

This chapter briefly reviews DGCNN and contrastive learning to understand the proposed pipeline.

3.1 DGCNN

DGCNN is an extension of the PointNet architecture, and it focuses on extracting semantic features from a point cloud by iteratively performing convolution on a dynamically updated neighborhood. Differently from graph CNNs, DGCNN utilizes EdgeConv (Edge Convolution) layers, which are dynamically updated after each layer of the network. EdgeConv focuses on addressing the same problems as Pointnet++ [17], which is unsuccessful on modelling the geometric relationships among points and capturing local features. However, this approach enhances PointNet's capabilities by constructing a local neighborhood graph and applying a convolution-like operation on the edge to connect the neighborhood pair of points. EdgeConv uses k Nearest neighbor algorithm to incorporate local neighborhood information instead of the farthest point sampling used in Pointnet++. This process facilitates the extraction of local features from the point cloud, making it suitable for semantic segmentation tasks.

Consider a F-dimensional point cloud with n points, denoted by $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^F$. In the simplest setting of $F = 3$, each point contains 3D coordinates $x_i = (x_i, y_i, z_i)$. An architecture for deep neural networks operates on the output of the previous layer at every layer, so the dimension F represents the feature dimensionality at a given layer. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents the local structure of the point cloud, where $\mathcal{V} = \{1, \ldots, n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are the vertices and edges, respectively. Constructed $\mathcal{G}$ as the k- NN graph in $\mathbb{R}^F$, containing directed edges of the form $(i, j_{i1}), \ldots, (i, j_{ik})$ such that points $x_{ji1}, \ldots, x_{jik}$ are the closest to x_i. Edge features are defined as $e_{ij} = h_\Theta(x_i, x_j)$, where $h_\Theta: \mathbb{R}^F \times \mathbb{R}^F \to \mathbb{R}^{F'}$ is some parametric nonlinear function parameterized by the set of learnable parameters Θ, as visualized in Figure 6.
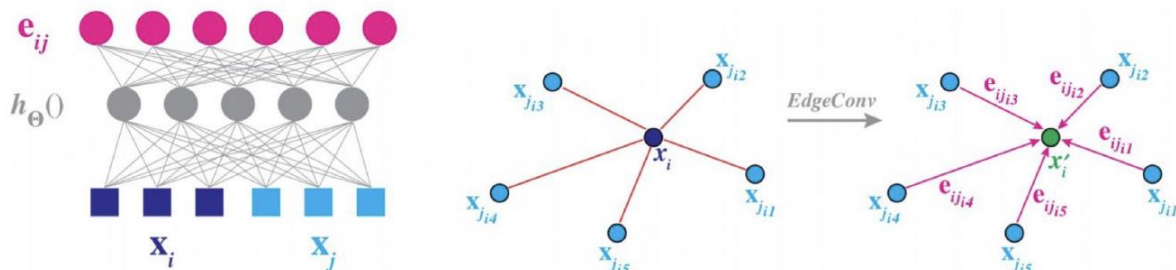
Figure 6: Left: An example of computing an edge feature, eij, from a point pair,xi and xj. Right: Visualization of the EdgeConv operation.

The DGCNN architecture is specifically designed for semantic segmentation tasks on point clouds. The segmentation model of DGCNN, as depicted in Figure 7, involves a series of three EdgeConv layers and three fully connected layers, producing a k-class score for each point. A max pooling operation is performed as a symmetric edge function, which makes the model permutation invariant while capturing global features. A spatial transformation module is used to align the input point cloud by applying a 3x3 matrix expected to be orthogonal and estimated during the training process. Each point is learned local geometric features using EdgeConv, which acts as MLP. The

goal of this study is to produce a semantic label for each point in a point cloud using DGCNN for semantic segmentation.

(a) The DGCNN semantic segmentation network architecture

(b) Spatial transformation block

(c) EdgeConv block
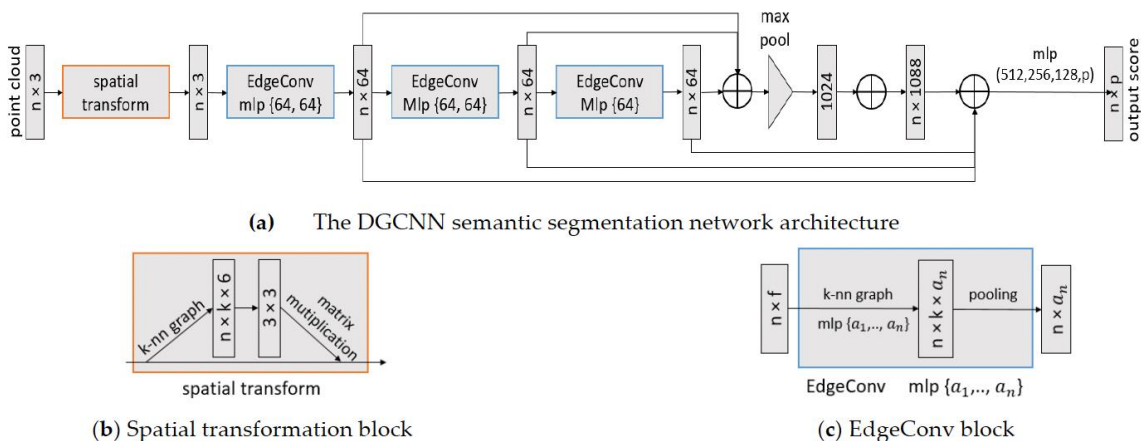
Figure 7: The DGCNN components for semantic segmentation architecture.

3.2 Self-Supervised Contrastive Learning

In this thesis, we utilize a self-supervised technique referred to as contrastive learning that aims to produce features that are distinguishable between unique inputs. The objective function encourages the outputs of a model, given similar input data points such as positive samples, to be close together, while it encourages the outputs produced from negative samples to be too far apart. Data points are defined as similar or dissimilar depending on the objective task to be trained. The points are subsequently fed through a model that learns to discriminate using a contrastive loss. A visualization of a typical contrastive learning framework is shown in Figure 8.

Figure 8: Visualization of a typical contrastive learning framework.
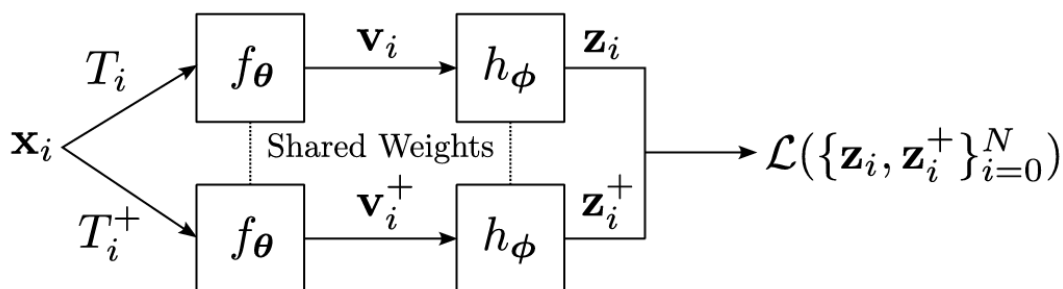
3.3 Loss Function

The goal is to maximize the similarity of $\mathbf{Z}^u$ and $\mathbf{Z}^v$ while minimizing the similarity with all the other projected vectors in the minibatch of graphs. We used the normalized temperature-scaled cross-entropy loss (NT-Xent loss) function in contrastive learning SimCLR [114]. The NT-Xent loss function for a positive pair of examples $\mathbf{Z}^u$ and $\mathbf{Z}^v$ is calculated as follows:

$$l(n, \mathbf{Z}^u, \mathbf{Z}^v) = -\log \frac{\exp(s(\mathbf{Z}_n^u, \mathbf{Z}_n^v)/\tau)}{\sum_{\substack{k=1 \\ k \neq n}}^{N} \exp(s(\mathbf{Z}_n^u, \mathbf{Z}_k^u)/\tau) + \sum_{k=1}^{N} \exp(s(\mathbf{Z}_n^u, \mathbf{Z}_k^v)/\tau)},$$

$$(1)$$

CHAPTER 4: THE PROPOSED METHOD AND EXPERIMENTS

4.1 The Proposed Method

In this paper, we propose a new self-supervised representation learning method for LiDAR data to reduce the dependence of deep learning networks on data annotation. Our goal is to build a dual-scale contrastive learning network based on a mixture of graphs and points to enhance the accuracy of the feature extraction network.

In order to adapt to the contrastive learning network, we designed two types of data augmentation modules, including data augmentation network and segment augmentation network. Data augmentation network relies on graph-based methods. We extract class-agnostic segments from the point cloud and propose a contrastive loss to be applied over the extracted segments in the segment augmentation network. Simultaneously, we project the graph-level features to the point level and make the contrastive learning with the features obtained through the original point cloud at the point level.

We use the same backbone encoder, DGCNN, to extract representation vectors for the entire network. EdgeConv extracts features of the local shape of the point cloud while maintaining alignment invariance. The features are also rearranged using self-attention after EdgeConv to extract global features.

The NT-Xent loss function is used to compute overall objective. We then cover downstream models that leverage this feature extractor for 3D semantic segmentation. The overview of the proposed method is shown in Figure 9.
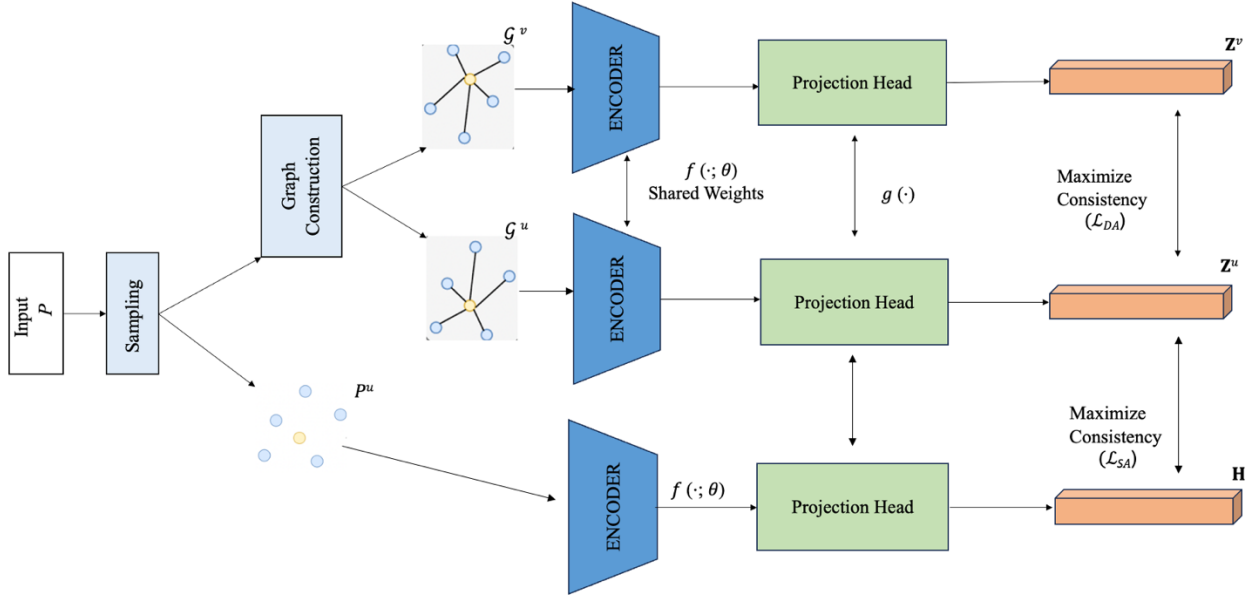
Figure 9: The overview of the proposed method.

4.1.1 *Graph Construction*

In a self-supervised setting, the data augmentation provides positive pairs of point clouds as input to the following networks. The augmentation module generates graphs via sampling and K-nearest neighbor as the data augmentation, illustrated in Figure 10. In the first step, given a point cloud P, we downsample the input point set to a subset of the input. Secondly, we perform random point sampling to generate two subsets P^u, P^v with a uniform scale. Then, the K-nearest neighbor adjacency graphs $\mathcal{G}^u$, $\mathcal{G}^v$ of the sampled point clouds as the augmented pairs of the input point cloud P are generated. Specifically, we search the spatial neighbors for each point in a point set P and then link them as a graph $\mathcal{G} = (V, E)$, where $V = 1, 2,\ldots,N$ symbolizes the set of vertices and $E \subseteq V \times V$ represents the set of edges. Each vertex i in graph $\mathcal{G}$ is associated with point pi in a point set P, which searches the K nearest spatial neighbors and connects with them to generate the edges.

Figure 10: An illustration of graph construction.

4.1.2 *Data Augmentation Network*

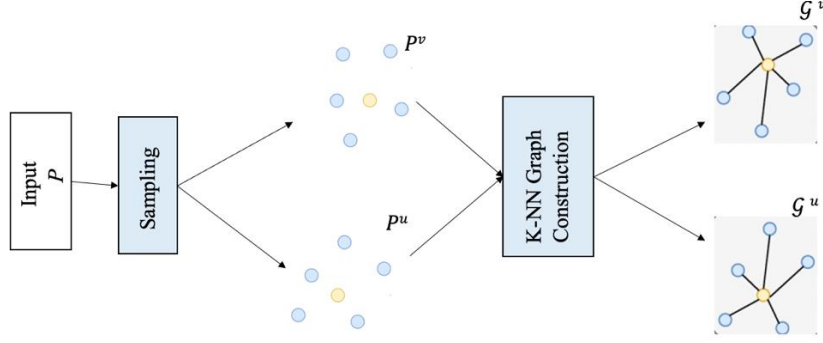Positive and negative samples are at the main of what makes comparison learning work. The data augmentation network enables a correspondence of representations by maximizing consistency between augmented graph pairs of the same point cloud via a contrastive loss in the latent space. We first obtained the graph level features by the feature extraction backbone, DGCNN encoder f $(\cdot; \theta)$. A neural network projection head $g(\cdot)$ is provided to map representations to the space where contrastive loss is applied. By leveraging the nonlinear transformation $g(\cdot)$, more information can be formed and maintained in the representation to prevent information loss induced by the contrastive loss. Besides, a nonlinear projection is better than a linear projection. Therefore, in this framework, we adopt MLP to obtain the projected vectors $\mathbf{Z}^u$ and $\mathbf{Z}^v$. We compute the cosine similarity between $\mathbf{Z}^u$ and $\mathbf{Z}^v$ as follows:

$$s(\mathbf{Z}^u, \mathbf{Z}^v) = \frac{\mathbf{Z}^{u\top}\mathbf{Z}^v}{\|\mathbf{Z}^u\| \, \|\mathbf{Z}^v\|}.$$

(2)

We compute the loss function l for the nth augmented pair of examples as:

$$l(n, \mathbf{Z}^u, \mathbf{Z}^v) = -\log \frac{\exp(s(\mathbf{Z}_n^u, \mathbf{Z}_n^v)/\tau)}{\sum_{\substack{k=1 \\ k \neq n}}^{N} \exp(s(\mathbf{Z}_n^u, \mathbf{Z}_k^u)/\tau) + \sum_{k=1}^{N} \exp(s(\mathbf{Z}_n^u, \mathbf{Z}_k^v)/\tau)},$$

(1)

where τ is the temperature parameter, N is the minibatch size. Our graph level discrimination loss function $\mathcal{L}_{DA}$ for a minibatch can be described as:

$$\mathcal{L}_{DA} = \frac{1}{2N} \sum_{n=1}^{N} [l(n, \mathbf{Z}^u, \mathbf{Z}^v) + l(n, \mathbf{Z}^v, \mathbf{Z}^u)].$$

(3)

4.1.3 *Segment Augmentation Network*

This is the approach taken by SegContrast [54], which demonstrated that a class-agnostic point cloud segmentation has been effective to segment the structures in the scene. Each step of self-supervised segment extraction is explained in Figure 11.
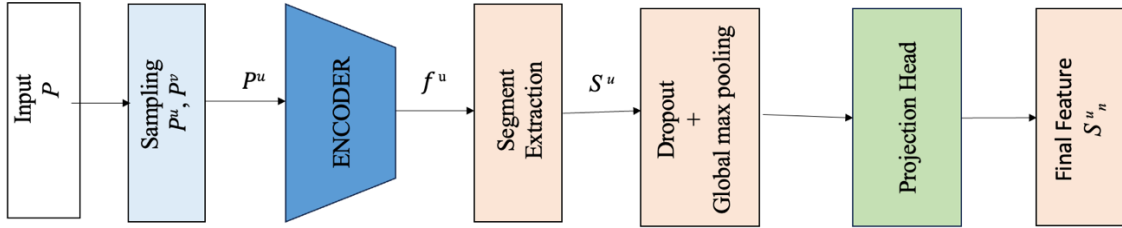
Figure 11: The overall architecture of the segment augmentation network.

Firstly, the point cloud P is used to extract class-agnostic segments S from it. We compute the point-wise features f^u by using DGCNN and determine the segment S with its point-wise feature using the point index of S^u from P^u. After dropout, global max pooling is applied over each segment, then S^u is projected to a latent space by a non-linear transformation $g\ (\cdot)$,

$$\mathbf{H} = g\ (S^u).$$

(4)

The same NT-Xent loss function is used to train the segment feature extraction network at the point level. We constructed a consistent contrastive strategy learning for both graph-level and point-level scales. We assume that for two different views of the same object, the features obtained by a robust feature extraction network should be the same.

The proposed method is then trained by minimizing NT-Xent loss to maximize the consistency of $\mathbf{Z}^u$ and $\mathbf{H}$. Then, same the loss function is calculated as,

$$l(n, \mathbf{Z}, \mathbf{H}) = -\log \frac{\exp(s(\mathbf{Z}_n, \mathbf{H}_n)/\tau)}{\sum_{\substack{k=1 \\ k \neq n}}^{N} \exp(s(\mathbf{Z}_n, \mathbf{Z}_k)/\tau) + \sum_{k=1}^{N} \exp(s(\mathbf{Z}_n, \mathbf{H}_k)/\tau)},$$

(5)

where τ is the temperature parameter. The final loss function $\mathcal{L}_{SA}$ is computed across all positive pairs in the minibatch,

$$\mathcal{L}_{SA} = \frac{1}{2N} \sum_{n=1}^{N} [l(n, \mathbf{Z}, \mathbf{H}) + l(n, \mathbf{H}, \mathbf{Z})].$$

(6)

4.1.4 *Overall Objective*

Finally, we calculate the resultant loss function during training as the combination of $\mathcal{L}_{DA}$ and $\mathcal{L}_{SA}$, where $\mathcal{L}_{DA}$ represents the graph level feature consistency and $\mathcal{L}_{SA}$ represents segment contrastive loss at the point level.

$$\mathcal{L} = \mathcal{L}_{DA} + \mathcal{L}_{SA}$$

(7)

4.2 Experiments

For verification of our method's effectiveness, we use an indoor dataset that is commonly used to benchmark 3D contrastive learning [21], [24], [59], [63] together with an extra outdoor lidar dataset, including the S3DIS [26] and the SemanticKITTI [29], respectively.

4.2.1 *Datasets*

The S3DIS dataset has become a common data benchmark and evaluation metric for point cloud semantic segmentation. This dataset consists of three different buildings, which contain five large-scale indoor areas, including offices, conference rooms, and open spaces, covering a total of 6020 m2. Each point in the point cloud is indicated by a nine-dimension vector of XYZ, RGB, and

normalized location. Each point is also labeled with one of 13 semantic elements according to their attributes, such as indoor items and furniture.

SemanticKITTI contains roughly 25,000 laser scans of outdoor driving environments. It has 20 different classes with 3D semantic and instance labels. The scans are of a full rotation around the vehicle, going out to a range of about 20 meters, with the spatial resolution decreasing with distance. We use sequences 1-7 and 9-10 as our training set and sequence 8 as our validation set.

4.2.2 Evaluation Metrics

On S3DIS and SemanticKITTI data sets, three metrics, including per-class intersection over union (IoU) [115] and mean IoU (mIoU) of each class [115]. These metrics quantitatively evaluate the performance of our method. IoU evaluates per-class segmentation result, while mIoU is the intersection of the predicted and ground truth labels over their union, averaged segmentation result considering all semantic classes.

$$IoU_i = \frac{c_{ii}}{c_{ii} + \sum_{j \neq i} c_{ij} + \sum_{k \neq i} c_{ki}}$$

(8)

$$mIoU = \frac{\sum_{i=1}^{N} IoU_i}{N}$$

(9)

where C is an $N \times N$ confusion matrix of the segmentation result represents.

4.2.3 Pre-training Dataset

We pretrained our method by using 10% of the ScannetV2 dataset. In the ScannetV2 dataset, there are a total of 1513 acquisition scenes and 21 categories, with 1201 scenes for training and 312 scenes for testing. We select 100 scenes for the pretraining channel.

4.2.4 *Implementation Details*

Our model is built using the PyTorch [40] package. We use PyTorch using the Python language on a 64-bit Ubuntu system equipped with three NVIDIA TITAN X GPUs for all deep learning tasks, including data loading, model building, and backpropagation during training. We use the S3DIS and SemanticKITTI for learning the self-supervised representation. By augmenting each point cloud into two relevant graphs defined in Section 4.1, we generate positive pairs of point clouds as the input.

For the entire network, we used DGCNN as the feature extractor to reduce the parameter size and facilitate comparison with existing methods. In addition, the self-attention layer is added after the DGCNN to enhance the network's ability to get global information about the input scene. DGCNN and self-attention are combined to extract dual-scale features for the entire network. We employ a 2-layer MLP as the projection heads. The Adam optimizer is also used with an initial learning rate of 0.001 and a weight decay of $1{\times}10{-}4$.

To verify the effectiveness of our method, we compare against it three state-of-the-art baseline algorithms: PointContrast[58], SegContrast [54], and DepthContrast[52], using their official implementations for pretraining and data preprocessing.

CHAPTER 5: RESULTS AND DISCUSSION

5.1 Semantic Segmentation on S3DIS

S3DIS consists of six large-scale indoor areas from three different buildings for a total of 273 million points annotated with 13 classes. For indoor scene segmentation, we train our model on the full S3DIS dataset to test the effectiveness of the network. This experiment shows how our segmentation architecture generalizes to real indoor scenes.
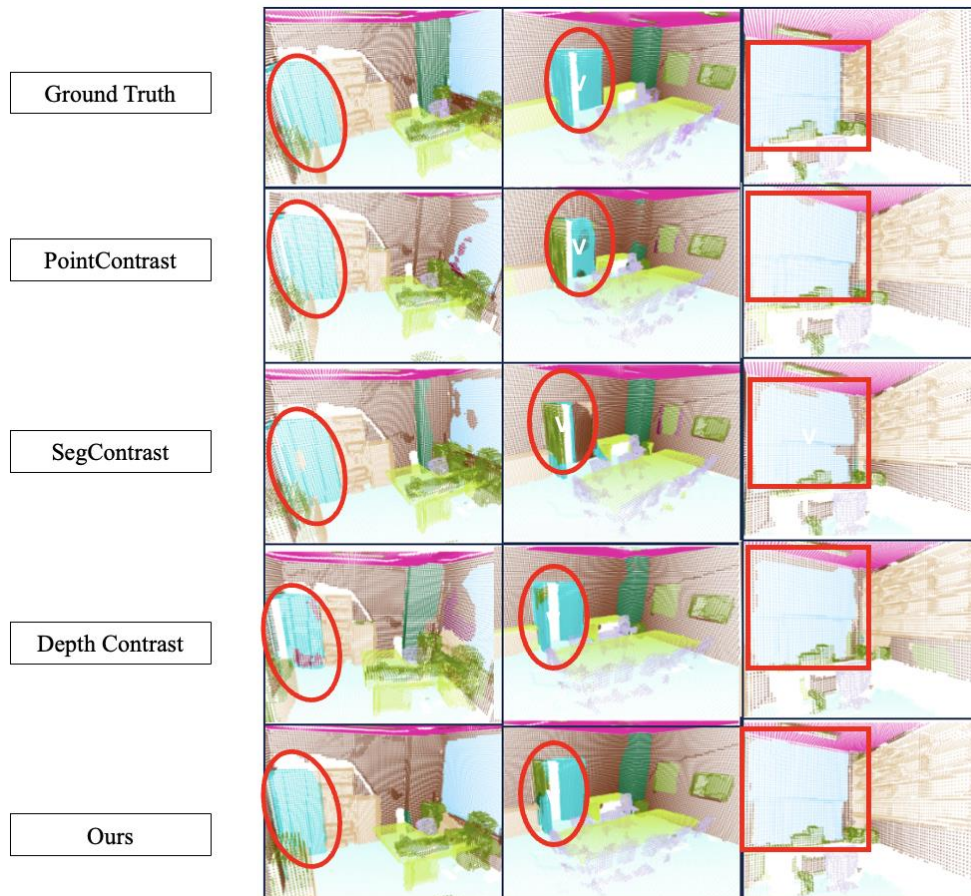

Figure 12: Visual comparison of semantic segmentation on the S3DIS dataset.

Figure 12 demonstrates the results of the semantic segmentation for three representative scenes, offering a visual representation of the impact of our proposed method. We colorize scenes from S3DIS with their semantic labels. Compared to other methods, our proposed method shows more

accurate segmentation result for most areas. Our method generally shows the closest result to ground truth, especially for the boundaries of walls, windows, and objects. It is likely that these differences are of little practical significance in the real world, which further supports our method's superiority.

Table 9: S3DIS semantic segmentation results (mIOU).

Method	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter	mIOU
PointContrast	91.68	96.59	83.21	0	40.38	58.12	70.42	76.77	87.13	69.33	68.77	82 .34	58.45	67.94
SegContrast	90.8	96.57	82.52	0	30.94	56.94	68.71	75.12	88.48	71.7	72.17	73.78	56.35	66.80
DepthContrast	90.99	95.45	80.99	0	32.52	51.94	61.57	74.55	87.28	71.62	71.34	67.71	57.85	64.91
Ours	91.81	96.44	84.11	0	40.14	58.64	81.53	77.63	89.82	83.74	76.03	84.5	59.45	70.22

Table 9 shows the segmentation results of four different methods per class and the average of all semantic classes to quantitatively assess the accuracy and quality of our segmentation model. Our method achieves the best mIoU with 70.22 % more than other competitive methods, e.g., PointContrast, SegContrast, and DepthContrast.

Table 10: Performance of representative methods on semantic segmentation using S3DIS (Area 5).

Method	Type	mIoU
Supervised (PointNet)	Point-based	47.7
Supervised (DGCNN)	Graph-based	56.1
Supervised (SPG)	Graph-based	62.1
OcCo	Alignment	49.5
Point-MAE	Reconstruction	61.0
3D jigsaw	Reconstruction	48.2
MaskSurf	Reconstruction	61.6
PointContrast	Contrast	70.9
CSC	Contrast	73.8
DepthConstrast	Contrast	64.8
SegContrast	Contrast	63.7
Multi-view rendering	Alignment	49.9
Ours	Contrast	71.4

Also, according to common research convention, Area-5 in the S3DIS data set is a test scene to better measure the generalization ability of our method. There are some differences observed in the objects present within Area 5 in contrast to those in other areas. Table 10 shows the quantitative evaluations of the experimental results. The proposed method achieves the best mIoU compared to other competitive self-supervised methods and supervised methods such as PointNet [16], SPG [51], and DGCNN [21], excluding CSC.

5.2 Semantic Segmentation on SemanticKITTI

We observe that our method is on the SemanticKITTI datasets. Using the model pre-trained on ScanNet shows performance improvements on SemanticKITTI as well, even though the datasets are quite different. Our method, when applied to SemanticKITTI, however, falls short of all other algorithms.
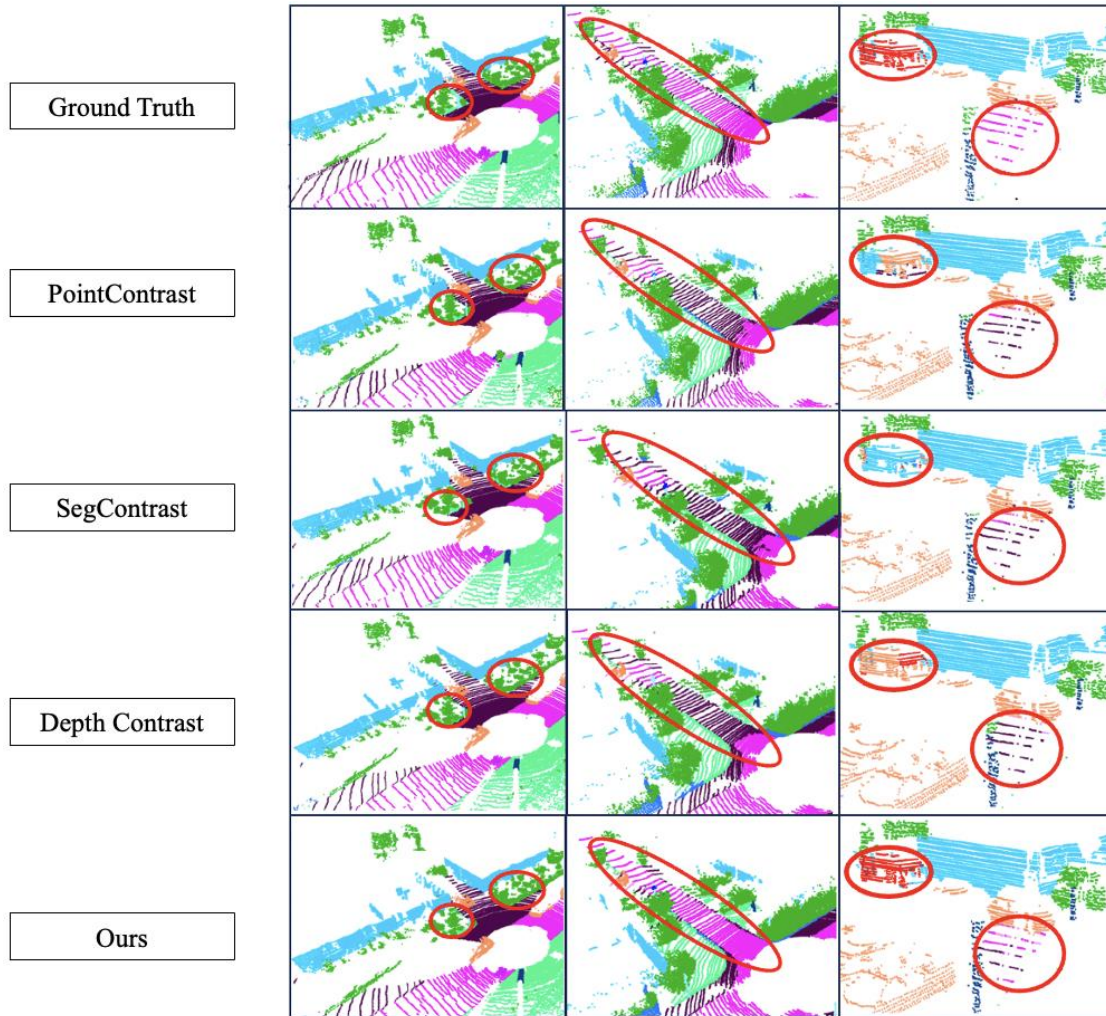
Figure 13: Visual comparison of semantic segmentation on the SemanticKITTI dataset.

Figure 13 shows a comparison of the semantic segmentation results of our method on the SemanticKITTI data set. Compared with ground truth, our self-supervised pretraining method produced more accurate segmentation results compared to other contrastive learning approaches. The segmentation results from our method contain more precise boundaries and finer details. Specifically, it achieved better performance at distinguishing the division between different structures, such as sidewalk and road, as shown by solid red circles.

Table 11: Per-class performance on SemanticKITTI using 0.1% of the annotated scans for fine-tuning.

Method	car	road	sidewalk	building	fence	vegetation	terrain	parking	pole	traffic sign	mloU
PointContrast	82.61	74.32	52.08	60.12	21.29	83.13	68.20	9	30.09	35.6	32.40
SegContrast	88.60	69.31	48.57	81.67	22.60	83.16	67.21	14.02	48.69	32.9	33.70
DepthContrast	89.11	83.34	64.09	83.43	23.63	83.43	70.57	20.61	46.66	37.3	35.71
Ours	92.20	85.34	66.09	87.70	25.43	85.00	72.57	22.61	48.67	40.4	37.90

The quantitative evaluations of the experimental results are provided in Table 11. It is possible to see that our method performs better in the per-class IoU, outperforming previous approaches in most of the classes. This evaluation shows that our approach performs better compared to the other contrastive learning methods, being able to better describe the different structures in the point cloud. Specifically, as displayed in Table 11, our method achieves a superiority of +2.2 mIoU over the best result in other methods.

Table 12: Performance on SemanticKITTI using different the annotated scans for fine-tuning.

Method	0.1%	1%	10%	100%
PointContrast	32.40	38.57	37.21	55.14
SegContrast	33.70	42.67	42.59	46.88
DepthContrast	35.71	47.06	53.48	56.34
Ours	37.90	51.54	54.25	58.18

Table 12 shows the results obtained by fine-tuning with different percentages of training data. Our method outperforms the other self-supervised approach.

For both indoor and outdoor datasets, our proposed model consistently outperforms others for all ratios, showing that our SSL model can effectively leverage the unlabeled data to improve the embedding features and the segmentation performance. Due to its powerful structured feature

learning capability, our method can learn to capture their discriminative features for semantic segmentation.

5.3 Ablations and Analysis

Table 13 presents the results of our ablation study, which investigates the contributions of our contrastive learning in the semantic segmentation model. By analyzing the results, we can verify the usefulness of each module in our proposed model.

5.3.1 *Pretraining Dataset*

The pre-training components were removed from the framework to verify its effectiveness, leaving only the dual-scale contrastive learning network structure. Table 12 shows segmentation accuracy of 60%, indicating that the network is generally effective, however the overall network accuracy is low, which demonstrates the effectiveness of the pre-training.

5.3.2 *Impact of the Graph Representation*

To evaluate the effect of the graph representation method, we design our model as point-based data augmentation and use PointNet as a backbone. We conduct experiment with this model on the S3DIS dataset and obtained 64.7% mIoU. When we compare it with the result of our proposed method, we see that graph representation and DGCNN increase the mIoU percentage.

5.3.3 *Self-attention Layer*

The widespread use of Transformers in computer vision has caused the attention mechanism to become more visible. Natural language processing and 2D image processing work often utilize attentional mechanisms for their powerful sequence modeling capabilities. However, to deal with the complexity and disorder of point cloud data, more Transformer-based point cloud processing networks have also been popular recently. The self-attention mechanism encodes the features of each point based on the position relationships between them. In our network, we employ the self-

attention to complement the lack of global modeling of the scene by the DGCNN feature extractor. We remove self-attention layer for the corresponding ablation studies to confirm the efficacy of this module. As illustrated in Table 13, the network lacking a self-attention layer shows a segmentation mIoU of 67.8%.

5.3.4 *Self-supervised Segment Extraction*

Our approach exploits the characteristics of outdoor LiDAR data to extract class-agnostic segments and applies the contrastive loss over these segments. Without self-supervised segment extraction, our method reaches an mIoU of 68.3%. The experiment suggests that our approach with class-agnostic segments can learn a more robust feature representation on downstream tasks.

Table 13: Ablation Study on S3DIS.

Method	mIoU (%)
Ours (No pre-training)	60.0
Ours (No DGCNN + graph representation)	64.7
Ours (No self-attention)	67.8
Ours (self-supervised segment extraction)	68.3
Ours (Completed)	70.22

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, we present an innovative approach for learning deep representations of 3D point clouds via contrastive learning that leverages a self-supervised strategy for learning LiDAR point cloud semantic segmentation. Two contrastive loss functions are specially designed to improve capturing more discriminate embeddings. Our strategy is then evaluated on different datasets and compared with other state-of-the-art feature representation learning methods.

Our methodology effectively brings together valuable geometric information from graph-based representations converted point clouds. Local geometric correlations between the input and its neighbors supports the importance of leveraging geometric cues in scene understanding tasks, resulting in more accurate segmentations. In addition, adding a self-attention layer after EdgeConv allows to leverage global geometric information, thereby enhancing the results of LiDAR point cloud segmentation.

One of the contributions of this research is the development of a comprehensive framework that integrates a dual-scale contrastive learning process based on point-level and graph-level strategies. As a result, the overall performance of our contrastive framework enhances the accuracy of feature extraction.

Another significant achievement of this work is to include segment augmentation network. Segment augmentation network includes to extract the segment-wise points and features. This approach allowed us to boost segmentation performance by discriminating the segmented structures on the point cloud.

Based on our results, our approach enhances 3D semantic segmentation tasks to a notable superiority over existing methods. Although the research has made significant contributions, there are also areas that need to be explored further.

6.2 Future Directions and Research Limitations

Effective and efficient deep learning architectures are significant in 3D semantic segmentation tasks. Although the proposed model has achieved several significant accuracy and efficiency improvements, the real-time segmentation and detection tasks have not been achieved. Computation is problem related to contrastive learning methods. Our method can also be applied to other problems with appropriate transformations on the datasets to generalize better models with less computation. In future works, we intend to investigate these issues, such as 4D semantic segmentation.

Reference

[1]   M. Johnson-Roberson, J. Bohg, M. Björkman, and D. Kragic, "Attention-based active 3D point cloud segmentation," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei: IEEE, Oct. 2010, pp. 1165–1170. doi: 10.1109/IROS.2010.5649872.

[2]   M. Liu, "Robotic Online Path Planning on Point Cloud," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1217–1228, May 2016, doi: 10.1109/TCYB.2015.2430526.

[3]   U. Asif, M. Bennamoun, and F. A. Sohel, "RGB-D Object Recognition and Grasp Detection Using Hierarchical Cascaded Forests," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 547–564, Jun. 2017, doi: 10.1109/TRO.2016.2638453.

[4]   J. Chen, Y. K. Cho, and Z. Kira, "Multi-view Incremental Segmentation of 3D Point Clouds for Mobile Robots," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1240–1246, Apr. 2019, doi: 10.1109/LRA.2019.2894915.

[5]   A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI: IEEE, Jun. 2012, pp. 3354–3361. doi: 10.1109/CVPR.2012.6248074.

[6]   X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D Object Detection Network for Autonomous Driving," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 6526–6534. doi: 10.1109/CVPR.2017.691.

[7]   B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D Object Detection from Point Clouds," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 7652–7660. doi: 10.1109/CVPR.2018.00798.

[8]   "3D object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities".

[9]   Y. Zeng *et al.*, "RT3D: Real-Time 3-D Vehicle Detection in LiDAR Point Cloud for Autonomous Driving," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3434–3440, Oct. 2018, doi: 10.1109/LRA.2018.2852843.

[10]   B. H. Wang, W.-L. Chao, Y. Wang, B. Hariharan, K. Q. Weinberger, and M. Campbell, "LDLS: 3-D Object Segmentation Through Label Diffusion From 2-D Images," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2902–2909, Jul. 2019, doi: 10.1109/LRA.2019.2922582.

[11]　J. Bruna, P. Sprechmann, and Y. LeCun, "Super-Resolution with Deep Convolutional Sufficient Statistics." arXiv, Mar. 01, 2016. Accessed: Feb. 05, 2024. [Online]. Available: http://arxiv.org/abs/1511.05666

[12]　J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation".

[13]　G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning Deep 3D Representations at High Resolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 6620–6629. doi: 10.1109/CVPR.2017.701.

[14]　M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs." arXiv, Aug. 07, 2017. Accessed: Feb. 05, 2024. [Online]. Available: http://arxiv.org/abs/1703.09438

[15]　Y. Li, L. Ma, Z. Zhong, D. Cao, and J. Li, "TGNet: Geometric Graph CNN on 3-D Point Cloud Segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3588–3600, May 2020, doi: 10.1109/TGRS.2019.2958517.

[16]　R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 77–85. doi: 10.1109/CVPR.2017.16.

[17]　C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space".

[18]　Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution On X-Transformed Points".

[19]　L. Yi, H. Su, X. Guo, and L. Guibas, "SyncSpecCNN: Synchronized Spectral CNN for 3D Shape Segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 6584–6592. doi: 10.1109/CVPR.2017.697.

[20]　M. Simonovsky and N. Komodakis, "Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 29–38. doi: 10.1109/CVPR.2017.11.

[21]　Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019, doi: 10.1145/3326362.

[22] W. Shi and R. Rajkumar, "Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 1708–1716. doi: 10.1109/CVPR42600.2020.00178.

[23] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks." arXiv, Feb. 04, 2018. Accessed: Feb. 05, 2024. [Online]. Available: http://arxiv.org/abs/1710.10903

[24] B. Gao, Y. Pan, C. Li, S. Geng, and H. Zhao, "Are We Hungry for 3D LiDAR Data for Semantic Segmentation? A Survey of Datasets and Methods," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6063–6081, Jul. 2022, doi: 10.1109/TITS.2021.3076844.

[25] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3D.net: A new Large-scale Point Cloud Classification Benchmark." arXiv, Apr. 12, 2017. Accessed: Feb. 05, 2024. [Online]. Available: http://arxiv.org/abs/1704.03847

[26] I. Armeni *et al.*, "3D Semantic Parsing of Large-Scale Indoor Spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 1534–1543. doi: 10.1109/CVPR.2016.170.

[27] Q. Hu *et al.*, "RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 11105–11114. doi: 10.1109/CVPR42600.2020.01112.

[28] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panoptic Segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 9396–9405. doi: 10.1109/CVPR.2019.00963.

[29] J. Behley *et al.*, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 9296–9306. doi: 10.1109/ICCV.2019.00939.

[30] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao, "SemanticPOSS: A Point Cloud Dataset with Large Quantity of Dynamic Instances." arXiv, Feb. 21, 2020. Accessed: Feb. 05, 2024. [Online]. Available: http://arxiv.org/abs/2002.09147

[31]   H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal Quebec Canada: ACM, Jun. 2009, pp. 609–616. doi: 10.1145/1553374.1553453.

[32]   S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised Representation Learning by Predicting Image Rotations." arXiv, Mar. 20, 2018. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/1803.07728

[33]   H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal Quebec Canada: ACM, Jun. 2009, pp. 609–616. doi: 10.1145/1553374.1553453.

[34]   G. Csurka, "Domain Adaptation for Visual Applications: A Comprehensive Survey." arXiv, Mar. 30, 2017. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/1702.05374

[35]   Y. Zhang, J. Lin, R. Li, K. Jia, and L. Zhang, "Point-DAE: Denoising Autoencoders for Self-supervised Point Cloud Learning." arXiv, Nov. 30, 2023. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2211.06841

[36]   X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-BERT: Pre-training 3D Point Cloud Transformers with Masked Point Modeling," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 19291–19300. doi: 10.1109/CVPR52688.2022.01871.

[37]   Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked Autoencoders for Point Cloud Self-supervised Learning," in *Computer Vision – ECCV 2022*, vol. 13662, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., in Lecture Notes in Computer Science, vol. 13662. , Cham: Springer Nature Switzerland, 2022, pp. 604–621. doi: 10.1007/978-3-031-20086-1_35.

[38]   Y. Zhang, J. Lin, C. He, Y. Chen, K. Jia, and L. Zhang, "Masked Surfel Prediction for Self-Supervised Point Cloud Learning," Jul. 2022, Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2207.03111

[39]   C. Min, X. Xu, D. Zhao, L. Xiao, Y. Nie, and B. Dai, "Occupancy-MAE: Self-supervised Pre-training Large-scale LiDAR Point Clouds with Masked Occupancy Autoencoders." arXiv, Oct. 09, 2023. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2206.09900

[40] M. Xu, Y. Wang, Z. Zhou, H. Xu, and Y. Qiao, "CP-Net: Contour-Perturbed Reconstruction Network for Self-Supervised Point Cloud Learning." arXiv, Mar. 28, 2022. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2201.08215

[41] Y. Chen *et al.*, "Shape Self-Correction for Unsupervised Point Cloud Understanding," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 8362–8371. doi: 10.1109/ICCV48922.2021.00827.

[42] S. Garg and M. Chaudhary, "SeRP: Self-Supervised Representation Learning Using Perturbed Point Clouds." arXiv, Sep. 13, 2022. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2209.06067

[43] J. Sauder and B. Sievers, "Self-Supervised Deep Learning on Point Clouds by Reconstructing Space".

[44] I. Achituve, H. Maron, and G. Chechik, "Self-Supervised Learning for Domain Adaptation on Point Clouds," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA: IEEE, Jan. 2021, pp. 123–133. doi: 10.1109/WACV48630.2021.00017.

[45] C. Zhang, J. Shi, X. Deng, and Z. Wu, "Upsampling Autoencoder for Self-Supervised Point Cloud Learning." arXiv, Mar. 21, 2022. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2203.10768

[46] G. Hess, J. Jaxing, E. Svensson, D. Hagerman, C. Petersson, and L. Svensson, "Masked Autoencoder for Self-Supervised Pre-training on Lidar Point Clouds," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, Jan. 2023, pp. 350–359. doi: 10.1109/WACVW58289.2023.00039.

[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv, May 24, 2019. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/1810.04805

[48] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked Autoencoders for Point Cloud Self-supervised Learning," in *Computer Vision – ECCV 2022*, vol. 13662, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., in Lecture Notes in Computer Science, vol. 13662. , Cham: Springer Nature Switzerland, 2022, pp. 604–621. doi: 10.1007/978-3-031-20086-1_35.

[49] Y. Zhang, J. Lin, C. He, Y. Chen, K. Jia, and L. Zhang, "Masked Surfel Prediction for Self-Supervised Point Cloud Learning." arXiv, Jul. 07, 2022. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2207.03111

[50] C. Min, X. Xu, D. Zhao, L. Xiao, Y. Nie, and B. Dai, "Occupancy-MAE: Self-supervised Pre-training Large-scale LiDAR Point Clouds with Masked Occupancy Autoencoders," Oct. 2023, Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2206.09900

[51] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional ShapeContextNet for Point Cloud Recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 4606–4615. doi: 10.1109/CVPR.2018.00484.

[52] Y. Liu, L. Yi, S. Zhang, Q. Fan, T. Funkhouser, and H. Dong, "P4Contrast: Contrastive Learning with Pairs of Point-Pixel Pairs for RGB-D Scene Understanding." arXiv, Dec. 23, 2020. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2012.13089

[53] L. Li and M. Heizmann, "A Closer Look at Invariances in Self-supervised Pre-training for 3D Vision." arXiv, Jul. 13, 2022. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2207.04997

[54] L. Nunes, R. Marcuzzi, X. Chen, J. Behley, and C. Stachniss, "SegContrast: 3D Point Cloud Feature Representation Learning Through Self-Supervised Segment Discrimination," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2116–2123, Apr. 2022, doi: 10.1109/LRA.2022.3142440.

[55] Z. Lu, Y. Dai, W. Li, and Z. Su, "Joint data and feature augmentation for self-supervised representation learning on point clouds," *Graph. Models*, vol. 129, p. 101188, Oct. 2023, doi: 10.1016/j.gmod.2023.101188.

[56] D. Wang and Z.-X. Yang, "Self-Supervised Point Cloud Understanding via Mask Transformer and Contrastive Learning," *IEEE Robot. Autom. Lett.*, vol. 8, no. 1, pp. 184–191, Jan. 2023, doi: 10.1109/LRA.2022.3224370.

[57] K. Fu, P. Gao, S. Liu, R. Zhang, Y. Qiao, and M. Wang, "POS-BERT: Point Cloud One-Stage BERT Pre-Training." arXiv, Apr. 03, 2022. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2204.00989

[58] K. Fu, P. Gao, R. Zhang, H. Li, Y. Qiao, and M. Wang, "Distillation with Contrast is All You Need for Self-Supervised Point Cloud Representation Learning." arXiv, Feb. 08, 2022. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2202.04241

[59] X. Chen and K. He, "Exploring Simple Siamese Representation Learning," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 15745–15753. doi: 10.1109/CVPR46437.2021.01549.

[60]   G. Mei, X. Huang, J. Liu, J. Zhang, and Q. Wu, "Unsupervised Point Cloud Pre-Training Via Contrasting and Clustering," in *2022 IEEE International Conference on Image Processing (ICIP)*, Bordeaux, France: IEEE, Oct. 2022, pp. 66–70. doi: 10.1109/ICIP46576.2022.9897388.

[61]   Y. Chen, M. Nießner, and A. Dai, "4DContrast: Contrastive Learning with Dynamic Correspondences for 3D Scene Understanding." arXiv, Jul. 22, 2022. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2112.02990

[62]   B. Mersch, X. Chen, J. Behley, and C. Stachniss, "Self-supervised Point Cloud Prediction Using 3D Spatio-temporal Convolutional Networks".

[63]   A. Sanghi, "Info3D: Representation Learning on 3D Objects using Mutual Information Maximization and Contrastive Learning." arXiv, Aug. 22, 2020. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2006.02598

[64]   L. Jing, L. Zhang, and Y. Tian, "Self-supervised Feature Learning by Cross-modality and Cross-view Correspondences," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 1581–1891. doi: 10.1109/CVPRW53098.2021.00174.

[65]   R. Zhang, L. Wang, Y. Qiao, P. Gao, and H. Li, "Learning 3D Representations from 2D Pre-Trained Models via Image-to-Point Masked Autoencoders," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, Jun. 2023, pp. 21769–21780. doi: 10.1109/CVPR52729.2023.02085.

[66]   B. Tran, B.-S. Hua, A. T. Tran, and M. Hoai, "Self-supervised Learning with Multi-view Rendering for 3D Point Cloud Analysis," in *Computer Vision – ACCV 2022*, vol. 13841, L. Wang, J. Gall, T.-J. Chin, I. Sato, and R. Chellappa, Eds., in Lecture Notes in Computer Science, vol. 13841. , Cham: Springer Nature Switzerland, 2023, pp. 413–431. doi: 10.1007/978-3-031-26319-4_25.

[67]   R. Zhang *et al.*, "PointCLIP: Point Cloud Understanding by CLIP," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 8542–8552. doi: 10.1109/CVPR52688.2022.00836.

[68]   X. Zhu *et al.*, "PointCLIP V2: Prompting CLIP and GPT for Powerful 3D Open-world Learning." arXiv, Aug. 26, 2023. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2211.11682

[69]   Y. Zeng *et al.*, "CLIP ² : Contrastive Language-Image-Point Pretraining from Real-World Point Cloud Data," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, Jun. 2023, pp. 15244–15253. doi: 10.1109/CVPR52729.2023.01463.

[70]   L. Xue *et al.*, "ULIP: Learning a Unified Representation of Language, Images, and Point Clouds for 3D Understanding," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, Jun. 2023, pp. 1179–1189. doi: 10.1109/CVPR52729.2023.00120.

[71]   A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision".

[72]   X. Liu, C. R. Qi, and L. J. Guibas, "FlowNet3D: Learning Scene Flow in 3D Point Clouds".

[73]   H. Mittal, B. Okorn, and D. Held, "Just Go With the Flow: Self-Supervised Scene Flow Estimation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 11174–11182. doi: 10.1109/CVPR42600.2020.01119.

[74]   W. Wu, Z. Y. Wang, Z. Li, W. Liu, and L. Fuxin, "PointPWC-Net: Cost Volume on Point Clouds for (Self-)Supervised Scene Flow Estimation," in *Computer Vision – ECCV 2020*, vol. 12350, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., in Lecture Notes in Computer Science, vol. 12350. , Cham: Springer International Publishing, 2020, pp. 88–107. doi: 10.1007/978-3-030-58558-7_6.

[75]   R. Li, G. Lin, and L. Xie, "Self-Point-Flow: Self-Supervised Scene Flow Estimation from Point Clouds with Optimal Transport and Random Walk," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 15572–15581. doi: 10.1109/CVPR46437.2021.01532.

[76]   S. Andreas Baur, D. Josef Emmerichs, F. Moosmann, P. Pinggera, B. Ommer, and A. Geiger, "SLIM: Self-Supervised LiDAR Scene Flow and Motion Segmentation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 13106–13116. doi: 10.1109/ICCV48922.2021.01288.

[77]   Y. Kittenplon, Y. C. Eldar, and D. Raviv, "FlowStep3D: Model Unrolling for Self-Supervised Scene Flow Estimation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 4112–4121. doi: 10.1109/CVPR46437.2021.00410.

[78]   B. Ouyang and D. Raviv, "Occlusion Guided Self-supervised Scene Flow Estimation on 3D Point Clouds." arXiv, Oct. 16, 2021. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2104.04724

[79] R. Li, C. Zhang, G. Lin, Z. Wang, and C. Shen, "RigidFlow: Self-Supervised Scene Flow Learning on Point Clouds by Local Rigidity Prior," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 16938–16947. doi: 10.1109/CVPR52688.2022.01645.

[80] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and Deformable Convolution for Point Clouds," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 6410–6419. doi: 10.1109/ICCV.2019.00651.

[81] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep Convolutional Networks on 3D Point Clouds".

[82] H. Zhao, L. Jiang, J. Jia, P. H. S. Torr, and V. Koltun, "Point Transformer".

[83] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point Transformer V2: Grouped Vector Attention and Partition-based Pooling." arXiv, Oct. 12, 2022. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2210.05666

[84] H. Ran, J. Liu, and C. Wang, "Surface Representation for Point Clouds".

[85] G. Qian, Y. Li, H. Peng, and J. Mai, "PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies".

[86] Y. Zhang, J. Lin, C. He, Y. Chen, K. Jia, and L. Zhang, "Masked Surfel Prediction for Self-Supervised Point Cloud Learning," Jul. 2022, Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2207.03111

[87] D. Rethage, J. Wald, J. Sturm, N. Navab, and F. Tombari, "Fully-Convolutional Point Networks for Large-Scale Point Clouds," in *Computer Vision – ECCV 2018*, vol. 11208, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., in Lecture Notes in Computer Science, vol. 11208. , Cham: Springer International Publishing, 2018, pp. 625–640. doi: 10.1007/978-3-030-01225-0_37.

[88] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "SEGCloud: Semantic Segmentation of 3D Point Clouds," in *2017 International Conference on 3D Vision (3DV)*, Qingdao: IEEE, Oct. 2017, pp. 537–547. doi: 10.1109/3DV.2017.00067.

[89] F. Liu *et al.*, "3DCNN-DQN-RNN: A Deep Reinforcement Learning Framework for Semantic Parsing of Large-Scale 3D Point Clouds," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct. 2017, pp. 5679–5688. doi: 10.1109/ICCV.2017.605.

[90] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Niessner, "ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans," in *2018 IEEE/CVF Conference on Computer Vision*

and Pattern Recognition, Salt Lake City, UT: IEEE, Jun. 2018, pp. 4578–4587. doi: 10.1109/CVPR.2018.00481.

[91] H.-Y. Meng, L. Gao, Y.-K. Lai, and D. Manocha, "VV-Net: Voxel VAE Net With Group Convolutions for Point Cloud Segmentation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 8499–8507. doi: 10.1109/ICCV.2019.00859.

[92] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 4490–4499. doi: 10.1109/CVPR.2018.00472.

[93] G. Krispel, D. Schinagl, C. Fruhwirth-Reisinger, H. Possegger, and H. Bischof, "MAELi: Masked Autoencoder for Large-Scale LiDAR Point Clouds".

[94] H. Yang *et al.*, "GD-MAE: Generative Decoder for MAE Pre-Training on LiDAR Point Clouds," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, Jun. 2023, pp. 9403–9414. doi: 10.1109/CVPR52729.2023.00907.

[95] Jing Huang and Suya You, "Point cloud labeling using 3D Convolutional Neural Network," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, Cancun: IEEE, Dec. 2016, pp. 2670–2675. doi: 10.1109/ICPR.2016.7900038.

[96] H. Zhou *et al.*, "Cylinder3D: An Effective 3D Framework for Driving-scene LiDAR Semantic Segmentation." arXiv, Aug. 04, 2020. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2008.01550

[97] L. Landrieu and M. Simonovsky, "Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Jun. 2018, pp. 4558–4567. doi: 10.1109/CVPR.2018.00479.

[98] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim, "Self-Supervised Learning of Point Clouds via Orientation Estimation," in *2020 International Conference on 3D Vision (3DV)*, Fukuoka, Japan: IEEE, Nov. 2020, pp. 1018–1028. doi: 10.1109/3DV50981.2020.00112.

[99] M. Afham, I. Dissanayake, D. Dissanayake, A. Dharmasiri, K. Thilakarathna, and R. Rodrigo, "CrossPoint: Self-Supervised Cross-Modal Contrastive Learning for 3D Point Cloud Understanding," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 9892–9902. doi: 10.1109/CVPR52688.2022.00967.

[100] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph Attention Convolution for Point Cloud Semantic Segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 10288–10297. doi: 10.1109/CVPR.2019.01054.

[101] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs".

[102] Z.-H. Lin, S. Y. Huang, and Y.-C. F. Wang, "Learning of 3D Graph Convolution Networks for Point Cloud Analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021, doi: 10.1109/TPAMI.2021.3059758.

[103] H. Lei, N. Akhtar, and A. Mian, "Spherical Kernel for Efficient Graph Convolution on 3D Point Clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3664–3680, Oct. 2021, doi: 10.1109/TPAMI.2020.2983410.

[104] Y. He, H. Yu, X. Liu, Z. Yang, W. Sun, and A. Mian, "Deep Learning Based 3D Segmentation: A Survey." arXiv, Jul. 26, 2023. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2103.05423

[105] M. Feng, L. Zhang, X. Lin, S. Z. Gilani, and A. Mian, "Point attention network for semantic segmentation of 3D point clouds," *Pattern Recognit.*, vol. 107, p. 107446, Nov. 2020, doi: 10.1016/j.patcog.2020.107446.

[106] D. P. Singh and M. Yadav, "Deep learning-based semantic segmentation of three-dimensional point cloud: a comprehensive review," *Int. J. Remote Sens.*, vol. 45, no. 2, pp. 532–586, Jan. 2024, doi: 10.1080/01431161.2023.2297177.

[107] H. Zhou and D. Zhang, "Graph-In-Graph Convolutional Networks For Brain Disease Diagnosis," in *2021 IEEE International Conference on Image Processing (ICIP)*, Anchorage, AK, USA: IEEE, Sep. 2021, pp. 111–115. doi: 10.1109/ICIP42928.2021.9506259.

[108] C. Wang, B. Samari, and K. Siddiqi, "Local Spectral Graph Convolution for Point Set Feature Learning," in *Computer Vision – ECCV 2018*, vol. 11208, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., in Lecture Notes in Computer Science, vol. 11208. , Cham: Springer International Publishing, 2018, pp. 56–71. doi: 10.1007/978-3-030-01225-0_4.

[109] G. Li *et al.*, "DeepGCNs: Making GCNs Go as Deep as CNNs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 6923–6939, Jun. 2023, doi: 10.1109/TPAMI.2021.3074057.

[110] Z. Liang, M. Yang, L. Deng, C. Wang, and B. Wang, "Hierarchical Depthwise Graph Convolutional Neural Network for 3D Semantic Segmentation of Point Clouds," in *2019 International Conference on Robotics and*

Automation (ICRA), Montreal, QC, Canada: IEEE, May 2019, pp. 8152–8158. doi: 10.1109/ICRA.2019.8794052.

[111] Y. Li, L. Ma, Z. Zhong, D. Cao, and J. Li, "TGNet: Geometric Graph CNN on 3-D Point Cloud Segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3588–3600, May 2020, doi: 10.1109/TGRS.2019.2958517.

[112] N. Zhang, Z. Pan, T. H. Li, W. Gao, and G. Li, "Improving Graph Representation for Point Cloud Segmentation via Attentive Filtering," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, Jun. 2023, pp. 1244–1254. doi: 10.1109/CVPR52729.2023.00126.

[113] Y. Geng *et al.*, "3DGraphSeg: A Unified Graph Representation- Based Point Cloud Segmentation Framework for Full-Range High-Speed Railway Environments," *IEEE Trans. Ind. Inform.*, vol. 19, no. 12, pp. 11430–11443, Dec. 2023, doi: 10.1109/TII.2023.3246492.

[114] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations." arXiv, Jun. 30, 2020. Accessed: Feb. 06, 2024. [Online]. Available: http://arxiv.org/abs/2002.05709

[115] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 07, pp. 12993–13000, Apr. 2020, doi: 10.1609/aaai. v34i07.6999.