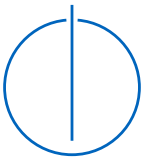
SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Exploiting Multi-Modality Context for Enhanced Online Adaptive Pseudo-Labeling of Point Clouds

Mert Kıray



SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

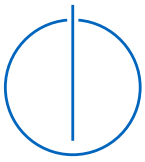
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Exploiting Multi-Modality Context for Enhanced Online Adaptive Pseudo-Labeling of Point Clouds

Ausnutzung des Multi-Modalen Kontexts für eine Verbesserte Adaptive Online Pseudoetikettierung von Punktwolken

Author:	Mert Kıray
Supervisor:	PD Dr. Ing. Habil. Federico Tombari
Advisor:	Lennart Bastian, Stefano Gasperini
Submission Date:	17.07.2023



I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 17.07.2023 Mert Kıray

Acknowledgments

First and foremost, I want to high-five two heroes of this journey - me, for pushing my limits, and my loyal companion, coffee, for being my lifeline through countless late nights. Fuelled by junk food and the quest for that sweet dopamine rush, we've formed an invincible team. I hope that years from now, I'll flip through these pages and smile at the memories.

A big, bear-sized hug goes out to my family. Your endless encouragement and unfaltering belief in me have been my guiding lights in even the darkest of times. Mom and Dad, your unwavering belief in me and constant encouragement have been my lighthouse in the storm. Your cheers at every minor success, and your soothing presence in times of stress have carried me through.

To my partner, Merve, my pillar of support and resilience, thank you. Through laughter and tears, triumphs and setbacks, you've stood by me, your unwavering support being my strength. We've braved the stormy seas together and have found our way to the shore, stronger and more unified than ever. Thank you for being my happy place.

And to my friends - the unsung heroes. Your rock-solid support, unwavering loyalty and the deep bonds of friendship that we've nurtured have been an integral part of this journey. You've been my strength, my solace.

My sincere gratitude goes to advisors Lennart and Stefano for their crucial guidance, and to my supervisor, Federico Tombari, for entrusting me with this significant opportunity.

This thesis is more than just a stack of papers. It's a testament to the collective effort, support, and love of my cherished ones. Each of you has contributed to this in your unique way, and for that, I am eternally grateful.

As I take my next steps, I recall the inspiring words of Mustafa Kemal Atatürk: "A good teacher is like a candle - it consumes itself to light the way for others." In this journey, you all have been my candles, consuming your time, energy, and resources to light my path. I am forever thankful for that. You all are the reason this journey was worth taking. Thank you.

Abstract

This thesis addresses the challenge of weakly supervised point cloud semantic segmentation by leveraging multi-modal information and introducing novel pseudo-labeling techniques. The objective is to reduce the laborious and time-consuming manual annotation process while maintaining competitive segmentation performance.

Existing state-of-the-art methods primarily focus on leveraging 3D modalities, such as point clouds and voxels, while disregarding the readily available 2D modality, including RGB images and depth maps. In contrast, this thesis proposes a comprehensive approach that integrates 2D RGB-D information into the pseudo-labeling and contrastive learning methods.

The proposed methodology exploits the geometric information derived from 2D-3D correspondences to establish consistency between the segmentation results of 2D and 3D modalities across the scene. To address the issue of sparse labels and enhance class representations, oversegmentation is employed to generate supervoxels and superpixels. The sparse labels are then propagated into the oversegmented regions, effectively increasing the label count. By matching supervoxel features with their corresponding superpixels in the embedding space, the proposed methodology enforces 2D-3D consistency throughout the scene. Furthermore, the sparse labels are leveraged to enforce consistency among supervoxels sharing the same label. Through the integration of 2D-3D consistency and contrastive learning, a robust online adaptive pseudo-labeling mechanism is introduced, eliminating the need for an additional network for pseudo-label generation.

Extensive experiments are conducted on popular datasets, including ScanNetv2 and 2D-3D-S, to validate the effectiveness of the proposed multi-modal integration, contrastive learning, and pseudo-labeling approaches. The experimental results demonstrate the superior performance and efficiency of the proposed methodology compared to existing methods, highlighting its potential for reducing manual annotation efforts and improving weakly supervised point cloud semantic segmentation.

Contents

1 Introduction

Semantic labeling of point clouds plays a crucial role in various applications, such as autonomous driving, robotics, and augmented reality. However, the manual annotation of point clouds is a labor-intensive and time-consuming process, which poses significant challenges in obtaining fully labeled datasets for training accurate segmentation models. To address this limitation, there has been a growing interest in leveraging weakly labeled or unlabeled data for point cloud semantic segmentation.

Recent advancements in weakly supervised 3D semantic segmentation have focused on techniques such as contrastive learning and pseudo-labeling. These approaches, such as Semantic Query Network (SQN) [1], One Thing One Click (OTOC) [2], and PointMatch [3], aim to achieve competitive segmentation results using a limited number of sparse labels. However, existing state of the art (SOTA) methods predominantly rely on 3D modalities, such as point clouds, while disregarding the available 2D modality, including RGB images and depth maps.

In this thesis, we propose a novel approach that integrates multi-modal information and introduces advanced pseudo-labeling techniques for weakly supervised point cloud semantic segmentation. Our methodology capitalizes on the geometric information derived from 2D-3D correspondences to establish consistency between the segmentation results of the 2D and 3D modalities. By incorporating the rich texture, color, and geometrical information provided by the 2D modality and combining it with the structural information present in the 3D point clouds, our approach aims to enhance the accuracy and comprehensiveness of point cloud segmentation.

The key contributions of this thesis can be summarized as follows:

- We propose a comprehensive framework that incorporates multi-modal information into the contrastive learning process. By leveraging oversegmentation, including the generation of supervoxels and superpixels, we address the issue of sparse labels and enhance class representations. This approach increases the label count and establishes consistency between supervoxels and superpixels by exploiting the geometric information derived from 2D-3D correspondences. These strategies result in a more robust and informative latent space construction, improving the overall performance of our contrastive framework.

- We incorporate the 2D modality, consisting of RGB images and depth maps, into

the pseudo-labeling process. This integration enriches the feature representation and captures fine-grained details in the scene. By leveraging the complementary information provided by RGB images and depth maps, our approach generates more accurate and reliable confidence pseudo-labels. These confidence pseudo-labels guide the learning process and enhance the segmentation performance of our approach.

- We introduce an online adaptive pseudo-labeling mechanism that dynamically adapts to the evolving model predictions. This mechanism eliminates the need for an additional network for pseudo-label generation, making the process more efficient and scalable.

- Extensive experiments are conducted on popular datasets, including ScanNetv2 [4] and 2D-3D-S [5], a superset of S3DIS [6], to evaluate the effectiveness of our proposed methodology. The experimental results demonstrate the superior performance and efficiency of our approach compared to existing methods.

This thesis contributes to the field of weakly supervised point cloud semantic segmentation by incorporating multi-modal information and introducing advanced pseudo-labeling techniques. Our proposed methodology addresses the limitations of existing methods and achieves competitive segmentation results while reducing the reliance on costly manual annotation. By leveraging the 2D and 3D modalities, our approach demonstrates the potential for efficient and accurate point cloud semantic segmentation in various real-world applications.

The outline of the rest of the thesis is as follows:

- Chapter 2 provides the necessary foundational information to understand the subsequent chapters.

- Chapter 3 presents an overview of the related work that has been conducted in the field.

- Chapter 4 outlines the methodology proposed in this thesis, covering preprocessing steps, feature extraction, contrastive learning, and pseudo-labeling.

- Chapter 5 presents a comprehensive comparison of our methodology with SOTA methods on various publicly available datasets, accompanied by detailed ablation studies that analyze and compare different components of our approach.

- Finally, Chapter 6 summarizes our findings, recaps the key contributions of the thesis, and proposes future research directions.

2 Background

In this Background chapter, we provide key information that will help readers understand the rest of the thesis. We aim to give a general overview, focusing on important concepts rather than detailed specifics. This foundational knowledge will assist readers in navigating and understanding the rest of our work.

2.1 Point Clouds

A point cloud is a collection of data points in a 3D coordinate system, typically defined by X, Y, and Z coordinates, representing the external surface of an object, while also potentially encapsulating attributes such as color, intensity, and surface normal details. An example point cloud can be seen in Figure 2.1.

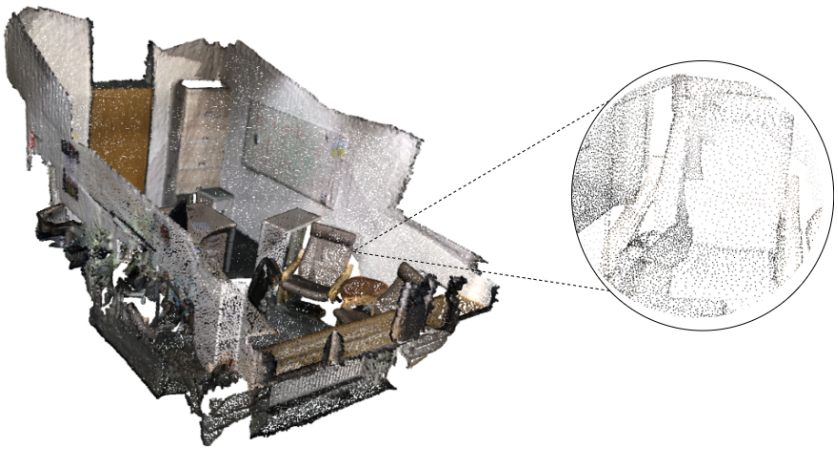


Figure 2.1: An example point cloud, which is a collection of 3D data points representing the external surface of objects.

In the past few years, high-quality 3D representations of the world, known as point clouds, have been obtained through expensive sensors such as a LiDAR; however, the emergence of affordable 3D sensors such as the Kinect sensor for the Microsoft Xbox 360 has revolutionized the accessibility of point clouds, making it possible for most future robots to perceive the world in 3D [7].

2.2 Semantic Segmentation

Semantic segmentation is a fundamental computer vision technique that plays a crucial role in understanding the content and structure of images or 3D scenes. By partitioning images or point clouds into coherent regions and assigning semantic labels to each region, semantic segmentation enables machines to recognize and differentiate between various objects or regions of interest. This fine-grained analysis allows for a comprehensive understanding of scene composition, facilitating tasks such as object detection, image understanding, and scene reconstruction.

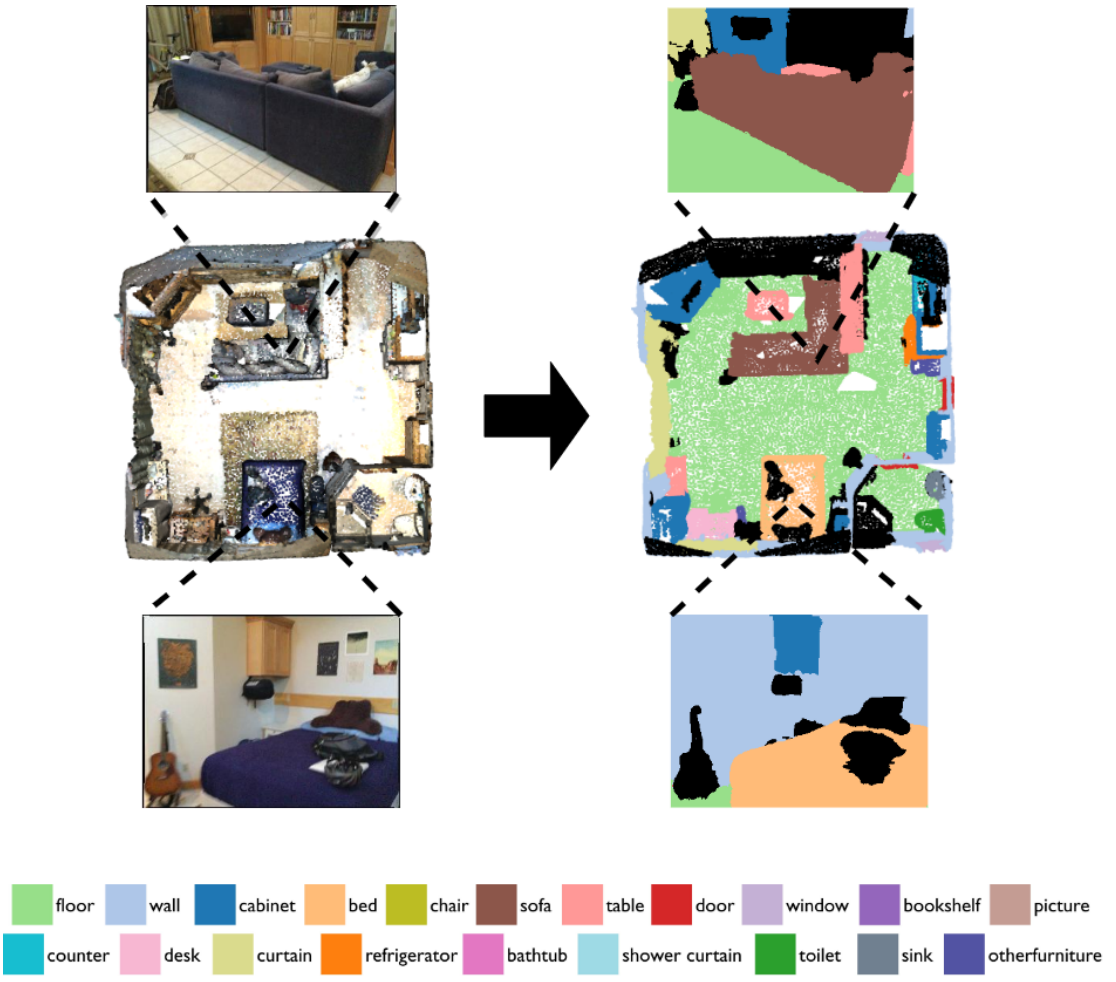


Figure 2.2: Semantic segmentation involves the task of assigning each pixel or point in a 2D image or 3D point cloud a semantic label.

As can be seen from Figure 2.2, in 2D semantic segmentation classifying pixels based on their semantic content provides a pixel-wise labeling of the image. On the other hand, 3D semantic segmentation extends this analysis to encompass three-dimensional data representations, such as point clouds. By associating semantic labels with individual points, 3D semantic segmentation allows for a detailed analysis of spatial environments, enabling applications in autonomous driving, robotics, virtual reality, and augmented reality.

Semantic segmentation has witnessed significant advancements in recent years, with the development of deep learning models that can effectively learn and extract meaningful features from images or point clouds. These models leverage Convolutional Neural Network (CNN) and other architectural variations to achieve SOTA performance in various semantic segmentation tasks.

2.3 Sparse Convolutions and Their Applications

CNN have been highly successful in various computer vision tasks, including semantic segmentation. However, their application to large-scale 3D data, such as point clouds, can be computationally demanding due to the inherent sparsity and irregularity of the data.

Sparse convolutions provide a solution for efficiently processing sparse data by selectively operating only on the non-zero or relevant elements, significantly reducing computational requirements. A general sparse tensor creation can be seen in Algorithm 1.

Algorithm 1 Sparse Tensor Creation

1: **procedure** CreateSparseTensor(Point Cloud Data)
2: Initialize an empty list of values X and coordinates $\mathcal{C}$
3: **for** each point p in the point cloud data **do**
4: Compute the value v and coordinate c for the point p
5: Append the value v to X and the coordinate c to $\mathcal{C}$
6: **end for**
7: **return** the sparse tensor X with coordinates $\mathcal{C}$
8: **end procedure**

Instead of applying convolutions across the entire input space, sparse convolutions target specific regions or points, enabling more efficient analysis of sparse data representations.

Sparse convolutions have found applications in various domains, including 3D

semantic segmentation. By exploiting the sparsity of point clouds or other 3D data representations, sparse convolutions allow for more efficient and accurate analysis of the data. These techniques enable the processing of large-scale point clouds without requiring dense voxelization or excessive memory consumption.

Furthermore, sparse convolutions facilitate the integration of contextual information from neighboring points or regions, enhancing the model's ability to capture long-range dependencies and improve semantic segmentation performance. The selective aggregation of information based on spatial proximity enables more effective analysis of local structures and global context within the sparse data.

Several architectures and models have been developed to leverage sparse convolutions for 3D semantic segmentation, such as Submanifold Sparse Convolutional Networks (SSCN) [8], and Minkowski Convolutional Neural Networks (MCNN) [9]. These models incorporate sparse convolutions as a fundamental building block, allowing them to efficiently process point clouds and achieve SOTA performance in 3D semantic segmentation tasks.

2.3.1 The Minkowski Engine

The Minkowski Engine [9] is a software framework specifically designed for efficient sparse tensor computations, particularly in the context of 3D data analysis. It provides a powerful tool for processing sparse data, such as point clouds, using sparse convolutions.

The Minkowski Engine [9] utilizes sparse tensors, which are data structures optimized for storing and manipulating sparse data. A sparse tensor consists of a set of non-zero values X along with their corresponding coordinates C as explained in Algorithm 1. In the context of 3D data, these coordinates typically represent the spatial locations of the non-zero values.

The Minkowski Engine [9] enables efficient sparse convolution operations by leveraging the sparse tensor representation. Let us consider a sparse tensor X with its corresponding coordinates $\mathcal{C}$. Sparse convolutions involve applying a set of learnable filters or kernels to the sparse tensor. Each filter is associated with its own weight parameters W.

The sparse convolution operation can be defined as follows:

$$\mathbf{x}_{\mathbf{u}} = \sum_{\mathbf{i} \in \mathcal{N}^D(\mathbf{u}, K, \mathcal{C}^{\text{in}})} W_{\mathbf{i}} \mathbf{x}_{\mathbf{i}+\mathbf{u}} \text{ for } \mathbf{u} \in \mathcal{C}^{\text{out}} \tag{2.1}$$

where K represents the kernel size, $\mathcal{C}^{\text{in}}$ is the predefined input coordinates of sparse tensor before the convolution operation, $\mathcal{C}^{\text{out}}$ is the predefined output coordinates of

sparse tensor after the convolution operation, and $\mathcal{N}^D(\mathbf{u}, K, \mathcal{C}^{\text{in}})$ is the set of offsets that are at most $\lceil \frac{1}{2}(K-1) \rceil$ away from u defined in $\mathcal{C}^{\text{in}}$ [9].

The Minkowski Engine [9] employs a mapping mechanism called kernel maps to represent how a sparse tensor is transformed into another sparse tensor using spatially local operations like convolution or pooling. For instance, in a 2D convolution with a kernel size of 3, a 3×3 convolution kernel consists of 9 weight matrices. Each kernel maps certain input coordinates to corresponding output coordinates. The mapping is represented as a pair of lists of integers: the in map ($\mathbf{I}$) and the out map ($\mathbf{O}$). In the in map, an integer $i \in \mathbf{I}$ indicates the row index of the coordinate matrix or feature matrix of an input sparse tensor. Similarly, in the out map, an integer $o \in \mathbf{O}$ indicates the row index of the coordinate matrix of an output sparse tensor. The elements in the lists are ordered in such a way that the k-th element i_k in the in map corresponds to the k-th element o_k in the out map. Thus, $(\mathbf{I} \rightarrow \mathbf{O})$ defines how the row indices of the input feature F_I map to the row indices of the output feature F_O. This mapping mechanism enables the Minkowski Engine [9] to efficiently perform operations on sparse tensors, optimizing computation and memory usage.

By selectively operating only on the non-zero elements of the sparse tensors and considering the neighboring coordinates, the Minkowski Engine [9] avoids unnecessary computations on empty regions and significantly improves computational efficiency.

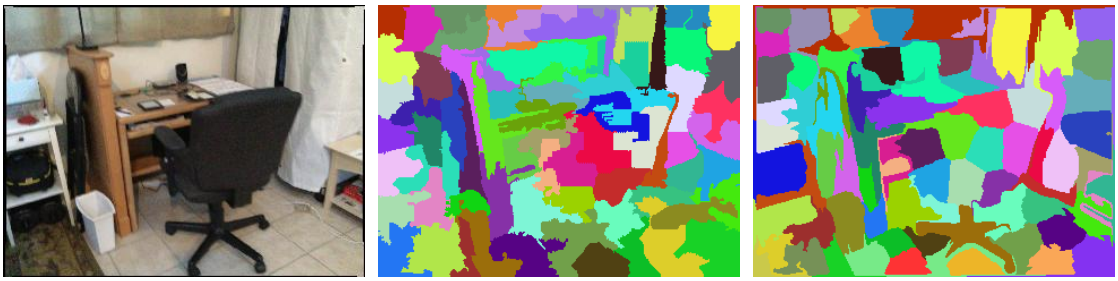
2.4 Oversegmentation

Oversegmentation is a widely used computational technique in the field of computer vision and image processing. It involves dividing an image or a region into smaller segments known as superpixels in 2D images and supervoxels in 3D point clouds. The main purpose of oversegmentation is to capture fine details and boundaries, providing a foundation for subsequent analysis tasks like object recognition, image editing, and scene understanding. By employing clustering and grouping algorithms based on various visual cues such as color, intensity, texture, spatial proximity, or even geometric features, oversegmentation generates a more detailed representation of the data. Importantly, this process is unsupervised, meaning it relies solely on intrinsic characteristics of the image or point cloud, without the need for manual annotation or extensive training. This intrinsic adaptability makes oversegmentation a valuable tool, especially in scenarios where labeled data is limited or acquiring annotations is prohibitively expensive.

2.4.1 Superpixels

Superpixels are compact and contiguous regions formed by grouping similar pixels together. They provide an intermediate representation that reduces the complexity of subsequent segmentation tasks. Superpixels preserve local spatial relationships and can be used to capture boundaries and structures more effectively than pixel-level analysis.

Various algorithms exist for generating superpixels, including simple and efficient approaches such as Simple Linear Iterative Clustering (SLIC) [10], which operates in the RGB color space and spatial coordinates. SLIC [10] partitions the image into compact regions by optimizing a distance metric that considers both color similarity and spatial proximity. Another popular algorithm is Superpixels Extracted via Energy-Driven Sampling (SEEDS) [11], which performs a hierarchical clustering of pixels based on color and texture information. Figure 2.3 shows the generated superpixels by different superpixel generation algorithms.



(a) 2D RGB Image (b) SEEDS (c) SLIC

Figure 2.3: Superpixels generated by different superpixel generation algorithms. (a) Shows the input 2D image. (b) shows the superpixels generated by SEEDS [11]. (c) shows the superpixels generated by SLIC [10].

2.4.2 Supervoxels

Supervoxels extend the concept of superpixels to three-dimensional space. They are employed in 3D scene analysis and segmentation tasks, where volumetric data, such as point clouds or voxel grids, are divided into coherent regions.

Similar to superpixels, supervoxels are compact and perceptually homogeneous regions that preserve local spatial relationships. They facilitate the extraction of meaningful 3D structures and boundaries while reducing the complexity of subsequent processing steps.

A notable work for generating supervoxels in an unsupervised fashion is Voxel Cloud Connectivity Segmentation (VCCS) [12] which uses a region growing variant of

k-means clustering. Also [13] proposed a supervised approach with graph-structured deep metric learning. Figure 2.4 shows oversegmentation of a point cloud to generate supervoxels.

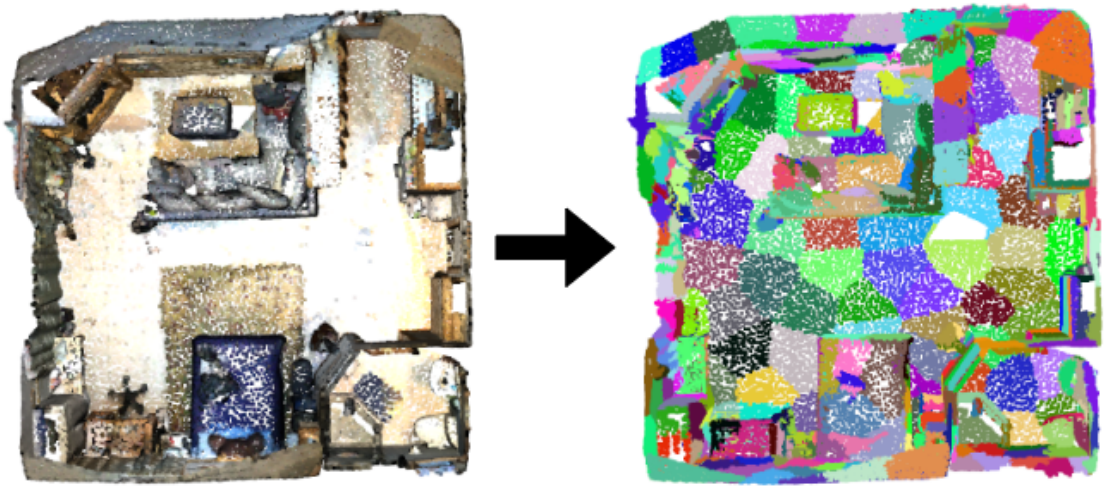


Figure 2.4: Illustration of the oversegmentation process in a point cloud to generate supervoxels. Supervoxels, as 3D extensions of superpixels, provide compact and perceptually homogeneous regions that preserve local spatial relationships, facilitating the extraction of meaningful 3D structures and boundaries.

2.5 Exploring the Principles of Contrastive Learning

Contrastive learning is an instrumental technique in self-supervised learning that strives to build representations by contrasting similar and dissimilar instances within a dataset. This methodology has received substantial attention in fields such as computer vision and natural language processing, due to its aptitude for learning distinctive and relevant features without explicit supervision.

2.5.1 An Introduction to Self-supervised Contrastive Learning

In the self-supervised contrastive learning paradigm, we construct positive and negative pairs of samples from an unlabeled dataset and aim to train a model to distinguish between them. As can be seen from Figure 2.5, the fundamental idea is to maximize the similarity within positive pairs, and concurrently minimize the similarity within negative pairs in the resulting feature space.

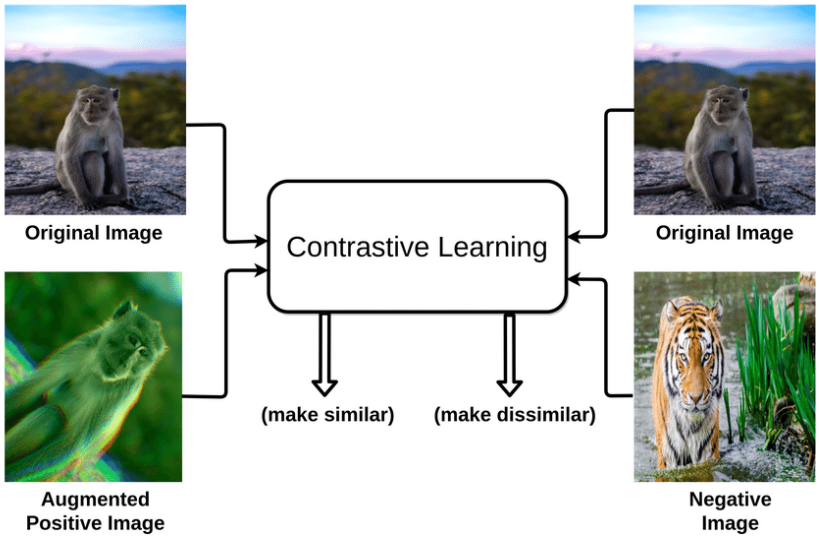


Figure 2.5: Overview of the self-supervised contrastive learning. The goal is to maximize the similarity of positive pairs and minimize the similarity of negative pairs in the latent space. Figure adapted from [14].

The contrastive loss function is frequently employed to train the model. This approach encourages the model to confer higher similarity scores to positive pairs and lower scores to negative pairs. Established methods such as SimCLR [15, 16] and MoCo [17, 18] leverage data augmentations to generate the positive and negative pairs, treating two differently augmented versions of the same image as a positive pair.

2.5.2 Transitioning to Supervised Contrastive Learning

Self-supervised contrastive loss assumes a lack of ground truth label information for generating positive and negative pairs. This is where the supervised contrastive learning approach, as proposed by Supervised Contrastive Learning (SupContrast) [19], diverges. Unlike SimCLR [15, 16] and MoCo[17, 18], SupContrast [19] proposes to utilize class labels to enhance the number of positive pairs. As can be seen from Figure 2.6, this is achieved by selecting augmented images from the same class as additional positive pairs, thereby making more efficient use of label information.

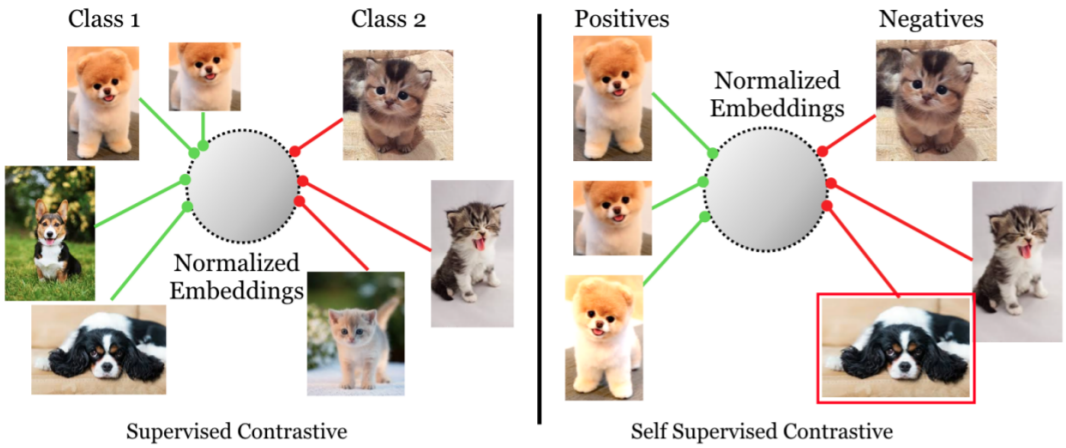


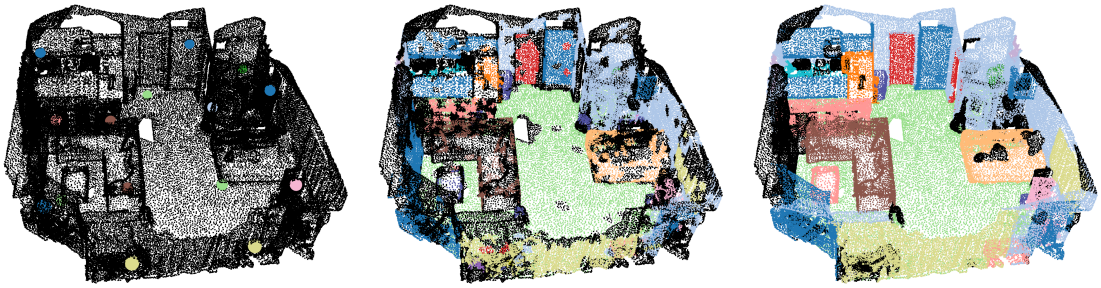
Figure 2.6: Overview of the supervised contrastive learning. This approach leverages class label information to increase the number of positive pairs by considering the class information. Figure adapted from [19].

2.6 The Concept and Application of Pseudo-Labeling

Pseudo-labeling is a powerful technique in machine learning that bridges the gap between supervised and unsupervised learning paradigms by leveraging unlabeled data to enhance model performance. In traditional supervised learning, models heavily rely on labeled data for training, which can be scarce or expensive to obtain. Pseudo-labeling tackles this challenge by harnessing the abundance of unlabeled data to augment the training process.

The concept of pseudo-labeling revolves around assigning labels to unlabeled data based on the predictions made by the model itself. During training, the model is initially trained on the available labeled data to learn from the ground truth labels. Subsequently, the trained model is applied to the unlabeled data, and class labels are assigned based on the highest predicted probabilities, forming what is known as pseudo-labels. The model is then fine-tuned using a combination of the original labeled data and the newly pseudo-labeled data.

The application of pseudo-labeling offers several advantages. It enables the utilization of large volumes of unlabeled data, which are often readily available in real-world scenarios. By leveraging this additional data, models can learn more representative and generalizable features, enhancing their ability to handle unseen samples. Moreover, pseudo-labeling provides a cost-effective solution by reducing the reliance on manual annotation, which can be time-consuming and costly.



(a) Sparse labels (b) Generated pseudo-labels (c) Ground truth labels

Figure 2.7: A visual illustration of the pseudo-labeling process.

However, pseudo-labeling also poses challenges. The quality of the pseudo-labels heavily relies on the accuracy of the model's predictions, and errors in pseudo-labeling can propagate during training, potentially leading to performance degradation. Thus, careful attention must be given to the confidence and reliability of the generated pseudo-labels to mitigate these issues.

Figure 2.7 provides a visual illustration depicting the impact of pseudo-labeling on a scene with initially sparse labeling. Figure 2.7a showcases a scene with limited sparse labels, indicating that only a small subset of data points has ground truth annotations. This sparse labeling scenario often poses challenges for models to fully comprehend the underlying patterns and relationships due to the scarcity of labeled guidance.

In contrast, Figure 2.7b shows the same scene after the application of pseudo-labeling, demonstrating how pseudo-labeling can increase the training data by assigning labels to previously unlabeled data points. These pseudo-labels are derived from the model's predictions on the unlabeled data, serving to augment the quantity and diversity of training data. The visual contrast between the sparsely labeled and pseudo-labeled scenes emphasizes the role of pseudo-labeling in expanding the training dataset and potentially enriching the quality of learned features.

3 Related Work

In this chapter, we look at important studies that have helped shape our work. We begin with an examination of the evolution of fully supervised semantic segmentation in both 2D and 3D, before moving on to an analysis of weakly supervised techniques. Furthermore, we study prior research that delves into the use of contrastive learning for point clouds.

3.1 Semantic Segmentation

3.1.1 Fully Supervised 2D Semantic Segmentation

Early works in semantic segmentation often relied on handcrafted features and classifiers. For instance, the TextonBoost [20] utilizes texture-layout filters that incorporate novel features derived from textons. These features enable the model to jointly capture both the patterns of texture and their spatial arrangement. [21] proposes a two-stage approach for figure/ground assignment in natural images using a conditional random field (CRF). [22] proposes a probabilistic model for labeling images into a predefined set of class labels using a generalization of the CRF approach. However, these methods were limited by the discriminative power of handcrafted features.

The Fully Convolutional Network (FCN) [23] proposes a method for semantic segmentation using fully convolutional networks that take input of arbitrary size and produce correspondingly-sized output with efficient training and inference. They adapt SOTA classification networks into fully convolutional networks by replacing the fully connected layers and transfer their learned representations by fine-tuning to the segmentation task.

An architecture that built upon the foundation laid by the FCN [23] is U-Net [24]. Initially developed for biomedical image segmentation, U-Net's [24] symmetrical contracting and expanding paths structure has found broad applicability due to its ability to localize features accurately.

Expanding on U-Net's [24] success, the ResU-Net [25] integrated residual connections from the ResNet [26], efficiently addressing the vanishing gradient problem and enhancing information flow throughout the network. This modification has made

ResU-Net [25] a potent tool for semantic segmentation, particularly when handling detailed input images or tasks requiring high-precision segmentation.

Deconvolution Network [27] introduces a novel semantic segmentation algorithm based on a learned deconvolution network. This network enables the generation of dense and precise object segmentation masks by progressively reconstructing object structures, overcoming limitations of fixed-size receptive fields and effectively handling object scale variations. Segnet [28] employs an encoder-decoder architecture with pooling indices, enabling efficient pixel-wise segmentation and accurate spatial reconstruction.

DeepLab variants [29, 30, 31] combines deep convolutional networks with atrous convolution and fully connected CRFs to achieve accurate and detailed segmentation results. The use of atrous convolution allows for the integration of larger context while maintaining computational efficiency, and the incorporation of fully connected CRFs further refines the segmentation boundaries for improved localization. PSPNet [32] incorporates a pyramid pooling module to capture multi-scale contextual information.

Attention mechanisms and transformer-based models have also been introduced to semantic segmentation tasks in recent years. PSANET [33] incorporates point-wise spatial attention to enhance scene parsing by selectively emphasizing informative image regions. Dual Attention Network [34], adaptively integrates local features with their global dependencies. Dense Prediction Transformer (DPT) [35] demonstrates the effectiveness of employing Vision Transformer (ViT) [36] backbone as opposed to the conventional CNN backbone.

3.1.2 Fully Supervised 3D Semantic Segmentation

For 3D semantic segmentation, the early pioneering methodologies were primarily focused on the voxelization of point clouds. VoxNet [37] employed a volumetric occupancy grid representation as their foundation and used a supervised 3D CNN. OctNet [38] introduced a better memory-efficient representation using octrees. Furthermore, O-CNN [39] leveraged a novel octree structure, enabling more efficient processing of 3D data with adaptive resolutions.

Simultaneously, point-based methods emerged to process raw 3D point clouds directly. PointNet [40] captured the unique structural characteristics of irregular and unordered point clouds. Building upon this work, PointNet++ [41] employed hierarchical learning to capture local structures within point clouds.

Many other models utilized the properties of graphs or introduced novel convolutions to process point clouds. For instance, DGCNN [42] leveraged the properties of graphs to capture local geometric structures and allowed the model to adapt to various scales and shapes. SpiderCNN [43] enhanced this by introducing parameterized convolutional

filters, while PointWeb [44] focused on enhancing local neighborhood features.

As the size of 3D datasets grew, there was a need for efficient semantic segmentation methods for large-scale point clouds. RandLA-Net [45] addressed this need by introducing an efficient neural architecture to directly extract per-point semantics for large-scale point clouds.

Innovative architectures, like Deep Parametric Continuous Convolutional Neural Networks [46], PointConv [47], and KPConv [48], introduce unique approaches to convolutional operations, enabling more effective processing of 3D point clouds and providing novel insights into handling their inherent irregularities. DualConvMesh-Net [49] introduced geodesic convolutions for processing 3D meshes.

Attention mechanisms were also integrated into 3D semantic segmentation through works like Graph Attention Convolution [50], which used graph attention mechanisms to enhance feature learning from point clouds. Hierarchical Point-Edge Interaction Network [51] and DeepGCNs [52] capitalized on the graph-like nature of point clouds for robust and flexible learning.

Moreover, several studies explored the use of sparse convolutions for efficient 3D segmentation. SSCN [8] and MCNN [9] are examples of this approach.

MCNN [9] presents a novel approach that utilizes sparse tensors and high-dimensional convolutions to address the inefficiencies of dense representations in 3D scans, where the majority of the space is empty. By representing non-empty space as coordinates and associated features using sparse tensors, the proposed approach achieves efficient storage and processing of high-dimensional data. The COO format is adopted for sparse tensors, enabling neighborhood queries and differentiation of points in different batches. Furthermore, the paper introduces a specialized library, Minkowksi Engine[9], for sparse tensors, which is further discussed in detail in the background section 2.3.1.

In numerous research studies, the utilization of 2D modalities has been explored to enhance the outcomes of 3D segmentation. In this regard, 3DMV [53] and Multi-view PointNet [54] employ backprojection techniques to fuse 2D features with the original 3D features. Virtual Multi-view Fusion [55] generates multiple virtual views of a scene, which are then fused with the 3D features.

Bidirectional Projection Network (BPNet) [56] leverages bidirectional projection module (BPM) to facilitate the interaction between complementary 2D and 3D information at various architectural levels. BPM is a key component of BPNet [56] for joint 2D and 3D scene understanding. The BPM is designed to enable bidirectional interaction between 2D and 3D visual domains by constructing skip connections between 2D and 3D sub-networks at the same decoder level. An overall architecture of BPNet [56] can be seen in Figure 3.1.

As can be seen from Figure 3.2, BPM first constructs a link matrix L between points and pixels according to the perspective projection from 3D to 2D space, given the 3D

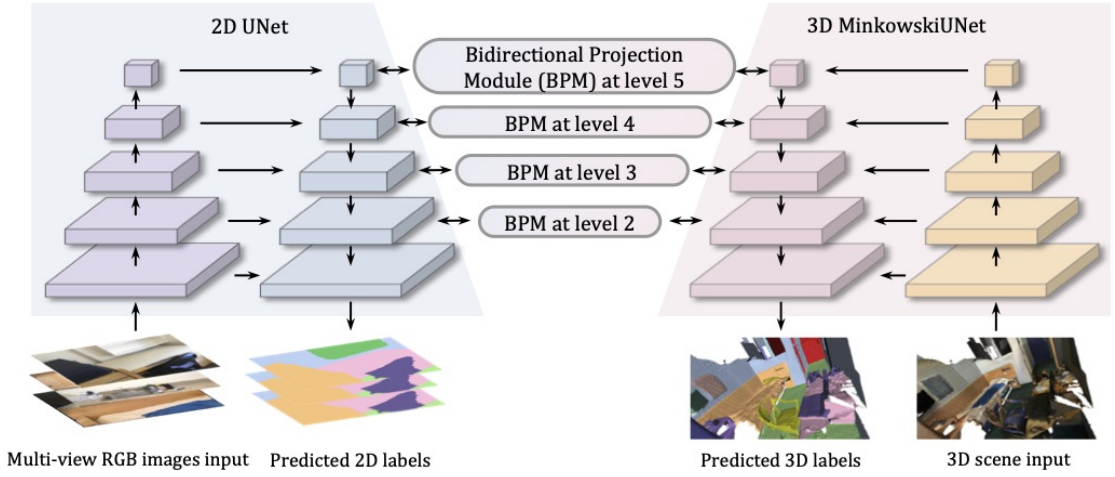


Figure 3.1: An overview of the BPNet [56] model architecture. The model leverages a Bidirectional Projection Module (BPM) to facilitate the interaction between 2D and 3D information at various architectural levels, improving joint 2D and 3D scene understanding. Figure adapted from [56].

scene and 2D image together with camera matrix M. Then, at multiple levels in the decoder, the BPM not only projects the 3D features to 2D space but also backprojects 2D features into 3D space according to the constructed link matrix. Finally, the projected features are concatenated with the original features followed by a 1×1 convolution to fuse them. This bidirectional interaction enables complementary information to flow between the 2D and 3D domains, improving joint 2D and 3D scene understanding [56].

In the context of a 3D scene with multiple 2D views, the projection of 3D features to each view can be achieved by utilizing the corresponding link matrix, as discussed. However, when it comes to the transformation of multi-view 2D features to 3D space, a fusion step becomes necessary after backprojection. While previous approaches like 3DMV [53] primarily rely on max-pooling for feature aggregation, BPNet [56] takes a different approach. Specifically, BPNet [56] leverage the power of two-layer sparse convolutions to effectively learn the impact factors associated with each view at every point. These learned impact factors are subsequently employed to perform a weighted sum of the backprojected features, facilitating a robust fusion of the multi-view 2D features within the 3D space. With these contributions, BPNet [56] is the most advanced and enhanced baseline to use for dealing with the fusion of 2D and 3D modalities while also leveraging the usage of sparse tensor and sparse convolutions effectively by using the Minkowski Engine[9]. For this reason, we utilized BPNet [56] as a backbone for extracting the features from the 2D and 3D modalities.

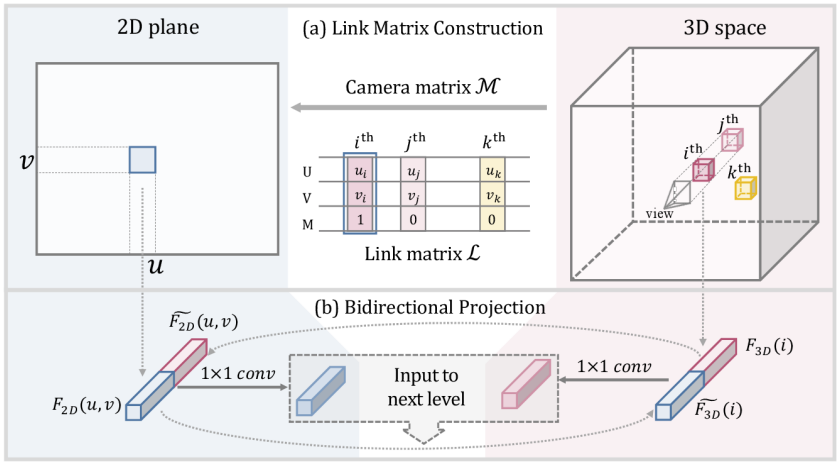


Figure 3.2: A detailed view of the Bidirectional Projection Module (BPM) used in the BPNet model. BPM projects 3D features to 2D space and backprojects 2D features into 3D space, enabling a bidirectional flow of information between the two domains. Figure adapted from [56].

Recent advancements have begun to explore joint tasks in 3D semantic segmentation. For instance, JSENet [57] proposed a joint semantic segmentation and edge detection network for 3D point clouds, while OCCUseg [58] introduced occupancy-aware 3D instance segmentation.

SOTA methods in 3D semantic segmentation include Swin3D [59], a pretrained Transformer backbone specifically designed for 3D indoor scene understanding, and Mix3D [60], a method that leverages out-of-context data augmentation to enhance 3D scene understanding.

3.1.3 Weakly Supervised 3D Semantic Segmentation

In recent years, weakly supervised 3D semantic segmentation has gained significant attention as a challenging problem in the field of computer vision. This area of research aims to tackle the task of semantic segmentation in 3D data using limited or weak forms of supervision. Several approaches have been proposed to address this challenge, leveraging different strategies and techniques.

One notable work in the field of weakly supervised 3D semantic segmentation is the SQN [1]. This paper presents a comprehensive framework that addresses the challenge of limited supervision by leveraging the assumption of semantic similarity between neighboring points in 3D space. The SQN [1] introduces a semantic query approach, where representations of neighboring points are queried and their semantic

similarity is considered to predict the final semantic labels. By incorporating this wider label propagation strategy, the SQN [1] enables the sparse training signals to be back-propagated to a larger spatial region, allowing for more comprehensive learning and improving the segmentation performance under weak supervision.

PointMatch [3] explores the utilization of unlabeled data for consistency training to enhance representation learning. This method adopts a scene-level augmentation technique, which generates multiple views of the same scene, and exploits the first augmented scene to generate pseudo-labels for subsequent scenes. By leveraging consistency training, PointMatch [3] achieves robustness and efficiency in representation learning by capitalizing on three key advantages. Firstly, the incorporation of various augmentations empowers the network to exhibit resilience against diverse perturbations on low-level input features. Secondly, the consistency target aids the model in extracting high-level semantic features directly from the point cloud data itself. Lastly, the self-training process operates as an implicit mechanism that propagates sparse training signals to unlabeled points, thereby facilitating the generation of dense pseudo-labels and enhancing the stability of the learning process. Moreover, PointMatch [3]introduces a gradual transition from supervoxel prior pseudo-labeling to point-wise pseudo-labeling during the training phase, adaptively adjusting the weight assigned to each technique as the model trains.

OTOC [2] introduces a self-training approach for weakly supervised 3D semantic segmentation. The method incorporates a graph propagation module and a relation network, working collaboratively to generate and propagate pseudo-labels in an iterative manner. This enables the model to learn from minimal supervision while leveraging low-level features such as color, coordinates, and 3D U-Net [24] features, along with predictions from the relation network, to propagate labels. The process involves the generation of supervoxels, efficient label propagation within the supervoxels, and the construction of a graph propagation module where each supervoxel represents a node in the graph. Updated pseudo-labels are generated using the low-level features and relation network predictions if the predicted pseudo-label criteria exceeds a given threshold.

Figure 3.3 provides an overview of the OTOC [2] model for weakly supervised 3D semantic segmentation, illustrating the iterative pseudo-label generation and graph propagation process.

In our proposed approach, similar to OTOC [2], we aim to perform weakly supervised semantic segmentation on point clouds by leveraging the utilization of oversegmented point clouds. However, our method diverges from OTOC [2] in that we exclusively employ the segmentation network for the generation and updating of pseudo-labels. We leverage the unlabeled data within the contrastive learning framework and integrate complementary 2D information as an additional check to address the uncertainty in

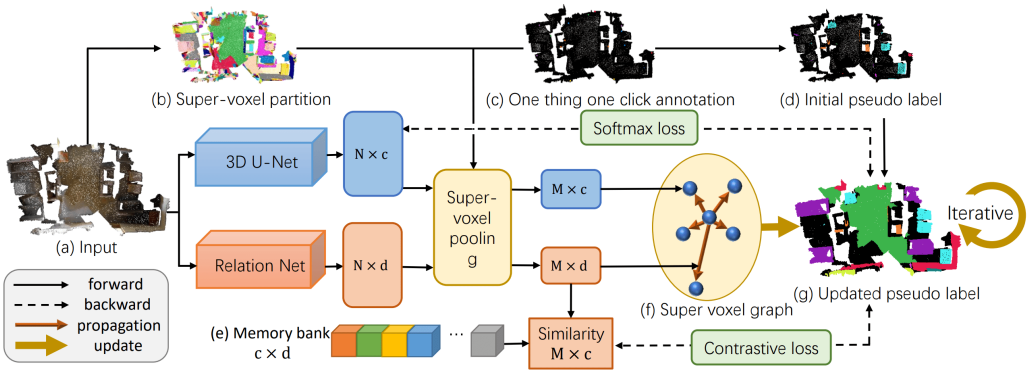


Figure 3.3: An overview of OTOC for weakly supervised 3D semantic segmentation. The model employs a graph propagation module and a relation network to iteratively generate and propagate pseudo-labels. Figure adapted from [2].

pseudo-labeling. Unlike OTOC [2], our approach simplifies the process by utilizing a single network for pseudo-label generation, resulting in increased efficiency in terms of the network parameters that need to be updated.

3.2 Contrastive Learning

Contrastive learning has shown great promise as a technique for unsupervised representation learning in the field of semantic segmentation. In this section, we explore several approaches that harness the power of contrastive learning to extract discriminative features and enhance the representation of point clouds specifically for the task of semantic segmentation.

The Fully Convolutional Geometric Features (FCGF) [61], revolutionizes feature extraction in point clouds by leveraging sparse tensors and sparse convolutions. Unlike traditional methods that require cropping or downsizing the point cloud, FCGF [61] enables direct feature extraction for each individual point. Additionally, FCGF [61] introduces point-level correspondence-based contrastive learning losses, which offer valuable insights for various downstream tasks that necessitate fine-grained analysis at the point level. These contrastive learning losses enhance the discriminative power of the learned features and facilitate their utilization in a wide range of applications.

PointContrast [62] extends the feature extraction capabilities of FCGF [61] by introducing the PointInfoNCE loss, which is specifically designed for point-level representations. This loss function is a variant of the well-established InfoNCE [63] loss widely used in contrastive learning. PointContrast [62] starts by generating two partial scans from

different views, ensuring a minimum overlap of 30%. To facilitate alignment, these partial scans are then transformed to the world frame. Positive pairs are constructed by selecting the same point from different partial scans, generated from distinct overlapping views. By employing the PointInfoNCE loss, PointContrast [62] pretrains the network by maximizing the agreement between positive pairs, thus promoting the learning of discriminative point-level features.

Superpixel-driven Lidar Representations (SLidR) [64] employs a knowledge distillation strategy to enhance feature similarity and consistency by leveraging both a pretrained 2D network and an untrained 3D network. The method initially generates superpixels using the SLIC [64] algorithm applied to the 2D images. Features are then extracted from the 2D images using a pretrained 2D network, while the 3D point cloud features are obtained using an untrained 3D network. Within each superpixel, feature averaging is performed. Additionally, correspondences between points within the superpixels and the point cloud are established. To align the features of the pretrained 2D network and the untrained 3D network, a superpixel-driven contrastive loss is employed. This process effectively distills the informative 2D feature information into the corresponding backprojected locations in the point cloud, facilitating enhanced feature similarity and consistent predictions.

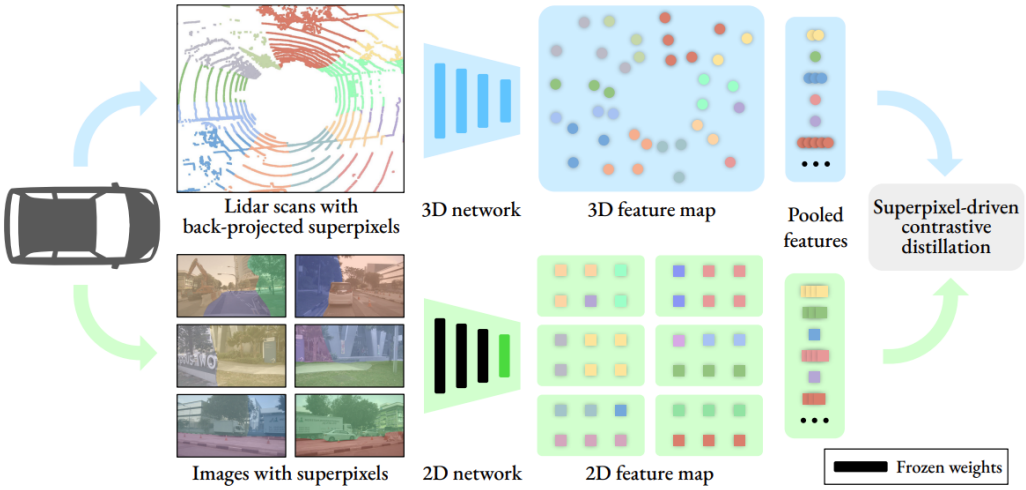


Figure 3.4: An overview of SLidR. The model employs a knowledge distillation strategy, leveraging a pretrained 2D network and an untrained 3D network to enhance feature similarity and consistency. Figure adapted from [64].

Figure 3.4 provides an overview of SLidR [64], illustrating the generation of superpixels, feature extraction, feature averaging, and the application of the superpixel-driven

contrastive loss to align the features of the pretrained 2D network and the untrained 3D network.

Pri3D [65], has emerged as a seminal work that has greatly influenced our research endeavors, particularly in the field of multi-view and point cloud consistency. Pri3D [65] puts forth a compelling proposition by harnessing the intrinsic characteristics of point clouds, characterized by their multi-view and multi-modality nature, to establish and enforce feature similarity between points in 3D space and their corresponding 2D pixels. To accomplish this objective, Pri3D [65] uses the PointInfoNCE [62] loss to promote feature similarity within both pixel-pixel and pixel-point correspondences. Through the integration of this innovative methodology, Pri3D [65] creates a consistent and stable prediction framework across different views and modalities.

Figure 3.5 serves as a visually informative representation of the framework of Pri3D [65], offering a comprehensive overview of the integrated approach that incorporates multi-view and multi-modality information to establish feature similarity between points and their corresponding 2D pixels.

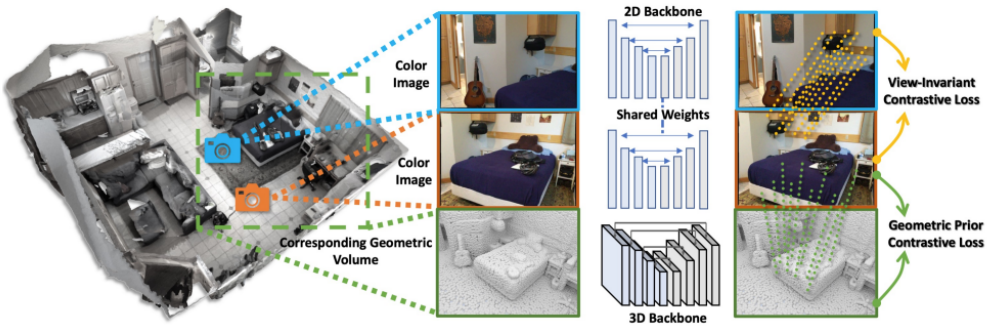


Figure 3.5: An overview of the Pri3D model. Pri3D leverages the multi-view and multi-modality nature of point clouds to establish feature similarity between points in 3D space and their corresponding 2D pixels. Figure adapted from [65].

The profound impact of Pri3D [65] on our research trajectory is evident in our deliberate integration of 2D-2D and 2D-3D correspondences within our research pipeline. Building upon the foundational insights provided by Pri3D [65], we recognize the inherent value in leveraging these correspondences as an additional criterion for pseudo-labeling, specifically by capitalizing on the agreement observed between the 2D-3D correspondences and their corresponding predictions. This novel integration enables us to exploit the rich and valuable insights offered by the 2D predictions, thereby complementing the conventional reliance on 3D prediction confidence. By

effectively utilizing the wealth of information available within both the 2D and 3D domains, our objective is to enhance the accuracy, robustness, and comprehensibility of the semantic segmentation of point clouds.

4 Methodology

Our methodology aims to address the challenges of weakly supervised semantic segmentation in point cloud data by integrating advanced techniques proposed in contrastive learning and pseudo-labeling. Our main motivation is to leverage the untapped potential of the 2D modality to enhance pseudo-labeling and improve the overall semantic segmentation performance.

In many prior works, the 2D modality is often overlooked, focusing solely on the 3D information for labeling and segmentation. However, we believe that incorporating the 2D counterpart can provide valuable insights and enhance the pseudo-labeling process. By analyzing and leveraging the complementary nature of the 2D and 3D modalities, we aim to improve the accuracy and robustness of weakly supervised semantic segmentation. An overview of our proposed model is illustrated in Figure 4.1.

To achieve our goals, we have designed a comprehensive methodology that encompasses various stages. The preprocessing stage plays a vital role in preparing the data by performing supervoxel oversegmentation, generating superpixels, and establishing 2D labels from the 3D point clouds. This enables us to capture both the spatial and visual context of the scenes.

To address the issue of sparse signal resulting from limited labeled data, we employ oversegmentation techniques to increase the signal count. By segmenting the point cloud into supervoxels and generating superpixels in the image plane, we propagate the sparse labels into the oversegmented regions, effectively increasing the supervision. The supervoxels and superpixels serve as key components in contrastive learning and pseudo-labeling stages.

To extract meaningful and discriminative features, we utilize a modified version of BPNet [56], enhancing it with additional convolutional layers to capture intricate patterns in both the 2D images and 3D point clouds. By extracting features from both modalities, we aim to leverage the unique information present in each modality and leveraging these features for contrastive learning and pseudo-labeling.

In the contrastive learning stage, we employ two different techniques. The first technique is inspired by Pri3D [65] and SLidR [64], utilizing unsupervised geometric loss to encourage similarity between superpixels and supervoxels. By extending this concept to the superpixel-supervoxel level, we aim to leverage the rich information provided by the 2D modality to enhance contrastive learning.

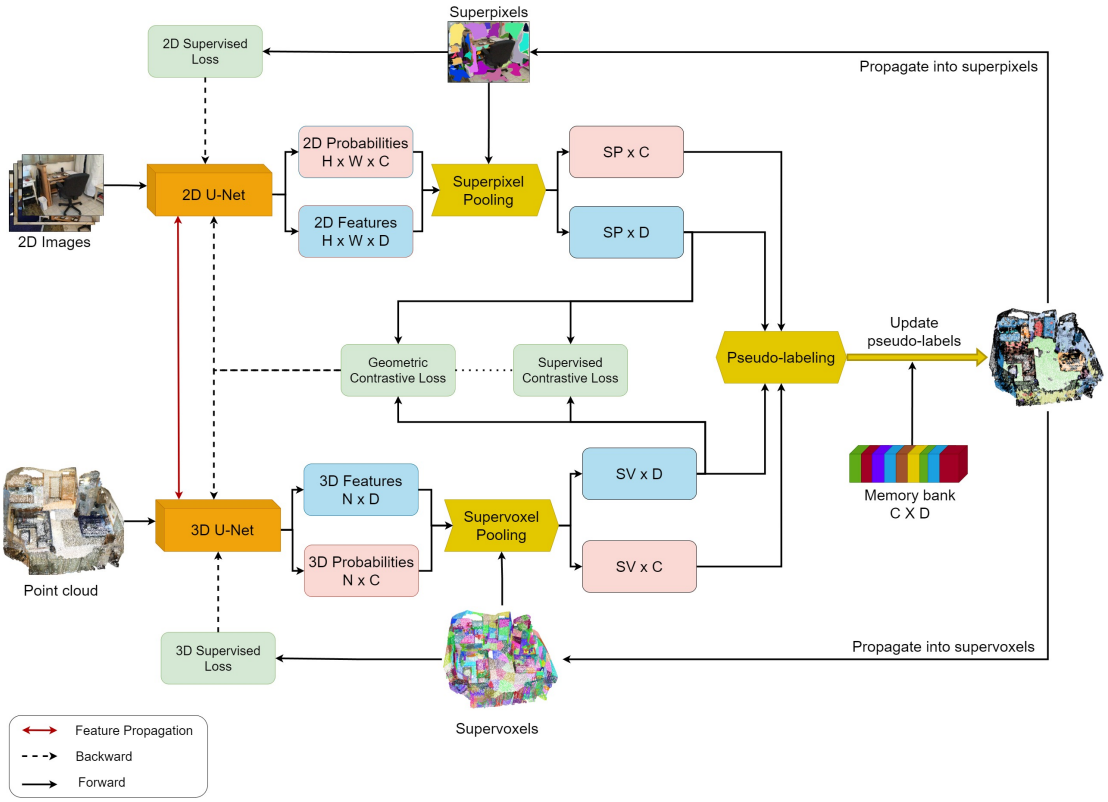


Figure 4.1: A general overview of our proposed model.

The second technique involves supervised contrastive loss, where we leverage known labels to encourage clustering of supervoxels with similar semantic information. By incorporating sparse labels and creating positive pairs based on known labels, we aim to improve the clustering of supervoxels, leading to more accurate and coherent segmentation results.

The final step in our methodology is pseudo-labeling, where we propagate labels based on prediction confidence and class prototypes. Here, we highlight the significance of integrating the 2D modality into the pseudo-labeling process. By leveraging the 2D information, we aim to enhance the accuracy of pseudo-labeling, resulting in improved semantic segmentation scores.

Throughout our methodology, we emphasize the importance of utilizing the 2D modality in conjunction with supervoxels to enhance pseudo-labeling and increase the overall performance of weakly supervised semantic segmentation in point clouds.

4.1 Preprocessing

In this section, we outline the preprocessing steps applied in our methodology.

4.1.1 Point Cloud Oversegmentation

To address the issue of sparse labels and improve the information utilization within the oversegmented regions, we employ oversegmentation techniques inspired by OTOC [2]. By leveraging oversegmentation, we enhance the signal derived from sparse labels and facilitate more efficient semantic segmentation and contrastive learning processes. Specifically, we assign labels directly to oversegmented supervoxels, rather than individual points.

To segment the point cloud into meaningful regions, we employ VCCS [12]. VCCS [12] utilizes voxelization and connectivity analysis to identify distinct regions within the point cloud, resulting in the generation of supervoxels. This process enables the partitioning of the point cloud into coherent and semantically meaningful regions.

4.1.2 Sparse Label Assignment within Supervoxels

To facilitate weakly supervised semantic segmentation, we assign sparse labels to the supervoxels based on the available annotated data. The sparse labels, representing specific object classes or semantic categories, are assigned to the supervoxels that contain corresponding points within their boundaries.

4.1.3 Superpixel Generation via Backprojection

Once the point cloud is oversegmented into supervoxels, we proceed to generate superpixels by backprojecting the supervoxels onto their corresponding 2D image counterparts. This involves projecting the 3D supervoxels onto their corresponding locations in the 2D image space. By aligning the supervoxels with their 2D representations, we create dense superpixel mappings. These mappings provide a contextual understanding of the underlying 3D structure, as can be seen in Figure 4.2.

Figure 4.2: Illustration of backprojecting 3D supervoxels onto the 2D image plane, generating superpixels that represent the context of the underlying 3D structure.

4.1.4 2D Image Label Generation via Backprojection

To establish a correspondence between the 2D images and the 3D point cloud data, we perform a backprojection process to generate 2D labels from the available 3D point cloud annotations. This involves projecting the sparse labels onto the corresponding camera poses using intrinsic and extrinsic matrices, resulting in 2D labels that align with the spatial locations of the objects or semantic categories within the point cloud. During the label assignment process, we incorporate occlusion masking by applying a 5 cm threshold, which ensures that the labels are accurately assigned, taking into account potential occlusions and improving the precision of the labeling process. It is important to note that the quality of these generated 2D labels may be lower compared to the original annotations, as the backprojection process does not account for certain filters and noise correction techniques applied in the original 2D labeling process.

4.2 Feature Extraction

In our methodology, we leverage the modified architecture of BPNet [56] to extract D-dimensional feature vectors from both the 2D images and point clouds using an added convolutional feature extraction head. By employing BPNet [56] with this modified feature extraction layer, we effectively harness the strengths of both modalities, capturing rich information from the 2D images and point clouds.

For supervoxels, which represent clusters of points in the 3D space, we compute the average feature vector by taking the mean of the feature vectors of all points within the supervoxel. This process captures the collective information of the points within each supervoxel, allowing us to represent the supervoxel with a single feature vector that encodes both local details and global contextual information. The computation of the average feature vector for a supervoxel can be represented as:

$$\mathbf{F}_{\text{supervoxel}} = \frac{1}{N_{\text{points}}} \sum_{j=1}^{N_{\text{points}}} \mathbf{F}_{\text{3D}}^{j} \tag{4.1}$$

where N_{points} is the total number of points in the supervoxel and $\mathbf{F}_{\text{3D}}^{j}$ represents the feature vector of the j-th point.

Similarly, for superpixels, which represent coherent regions in the 2D image space, we compute the average feature vector by taking the mean of the feature vectors of all pixels within the superpixel. This averaging operation allows us to summarize the information within each superpixel, capturing the overall characteristics and context of the region. The computation of the average feature vector for a superpixel can be represented as:

$$\mathbf{F}_{\text{superpixel}} = \frac{1}{N_{\text{pixels}}} \sum_{i=1}^{N_{\text{pixels}}} \mathbf{F}_{\text{2D}}^{i} \tag{4.2}$$

where N_{pixels} represents the total number of pixels in the superpixel and $\mathbf{F}_{\text{2D}}^{i}$ denotes the feature vector of the i-th pixel.

The resulting average pooled features from supervoxels and superpixels contain rich contextual information, capturing the collective characteristics of the points within each supervoxel and the pixels within each superpixel. These average feature vectors serve as more compact and representative descriptors of the supervoxels and superpixels. They preserve the essential information from the original feature vectors while reducing their dimensionality and computational complexity.

These average pooled features provide a holistic understanding of the scene, encapsulating both local and global cues, which is crucial for effective semantic segmentation.

Leveraging these features, we can better capture the relationships and contextual dependencies within the scene.

In our methodology, we utilize the average pooled features for two main purposes: contrastive learning and constructing class prototypes for pseudo-labeling. The average pooled features serve as the input for our contrastive learning techniques, allowing us to learn meaningful representations that capture the similarities and differences between supervoxels and superpixels. Additionally, these features are utilized to construct class prototypes, which are essential for the pseudo-labeling process.

4.3 Contrastive Learning

In our methodology, we integrate contrastive learning to address the limitations of sparse labels and enhance the quality of learned representations. Our contrastive learning pipeline enables the model to capture fine-grained differences and similarities among samples by leveraging both labeled and unlabeled data. We employ two distinct contrastive learning techniques: Unsupervised geometric contrastive learning and sparse label-aware supervised contrastive learning.

4.3.1 Unsupervised Geometric Contrastive Learning

In our approach, we leverage the established correspondence between superpixels and supervoxels to incorporate an unsupervised contrastive learning technique. Our goal is to enhance the similarity between the average pooled features of superpixels and supervoxels. This process is illustrated in Figure 4.3.

We utilized the PointInfoNCE [62] loss to measure the contrastive learning objective:

$$\mathcal{L}_{\text{geo}} = -\sum_{(i,j)\in\mathcal{P}} \log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k\in\mathcal{N}_{i,j}} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{4.3}$$

In this equation, (i, j) represents a positive pair of superpixel-supervoxel correspondences, where z_i and z_j denote the average pooled features of the superpixel and supervoxel, respectively. The term $\mathcal{P}$ represents the set of all positive pairs, and $\mathcal{N}_{i,j}$ denotes the set of negative pairs for the positive pair (i, j). The temperature parameter τ controls the sharpness of the contrastive loss function, with higher values focusing more on relative similarities and lower values emphasizing distinctions between similar and dissimilar pairs.

The similarity function $\text{sim}(z_i, z_j)$ computes the cosine similarity between the feature representations of z_i and z_j. It is defined as:

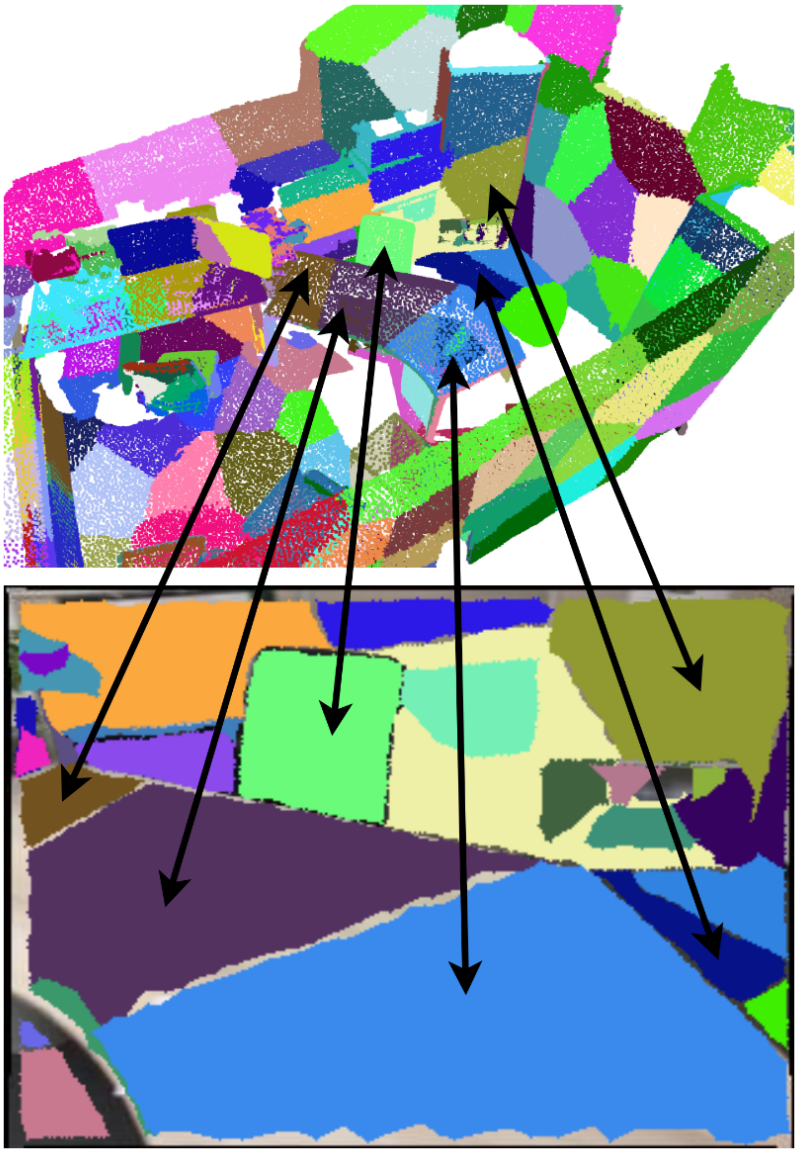


Figure 4.3: Illustration of the unsupervised geometric contrastive learning process. The process aims to increase the similarity between the average pooled features of corresponding superpixels and supervoxels in the latent space.

$$\text{sim}(z_i, z_j) = \frac{z_i \cdot z_j}{|z_i| \cdot |z_j|} \tag{4.4}$$

where z_i and z_j represent the feature vectors of the superpixel and supervoxel, respectively. The cosine similarity normalizes the similarity scores, making the contrastive loss invariant to the scale of the representations.

By using the unsupervised geometric contrastive learning we aim to bring the average pooled features of corresponding superpixels and supervoxels closer together in the latent space, encouraging the model to learn meaningful representations that capture the underlying similarities between them. This unsupervised geometric contrastive learning enables our model to discover and leverage the intrinsic geometric properties of the scene.

4.3.2 Sparse Label-Aware Supervised Contrastive Learning

In our methodology, we address the challenge of sparse labels by incorporating a supervised contrastive learning technique. The sparsity of labeled samples poses difficulties in training accurate and generalizable models. To overcome this limitation, we leverage the available sparse labels and enhance the discriminative power of the learned representations. An illustration of this process is shown in Figure 4.4.

By integrating supervised contrastive learning into our framework, we aim to encourage the clustering of supervoxels belonging to the same class while pushing apart supervoxels from different classes in the latent space.

For this purpose, we utilize the Supervised Contrastive Loss (SupCon) [19], which can be expressed as:

$$\mathcal{L}_{\text{sup}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\text{sim}(z_i, z_p)/\tau)}{\sum_{k \in N(i)} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{4.5}$$

In this equation, i represents an anchor supervoxel where I represents the set of all supervoxels, p denotes a positive supervoxel within the same class as the anchor, $P(i)$ represents the set of all supervoxels belonging to the same class as the anchor i, and $N(i)$ represents the set of all supervoxels from different classes than the anchor i. The index set I contains all anchor supervoxels.

The temperature parameter τ controls the sharpness of the contrastive loss function, similar to Equation 4.3. However, unlike the unsupervised case, we have a normalization factor $\frac{-1}{|P(i)|}$ in Equation 4.5.

The inclusion of the normalization factor $\frac{-1}{|P(i)|}$ in Equation 4.5 is essential in supervised contrastive learning as it balances the loss contributions across anchor supervoxels. By dividing the loss by the number of positive supervoxels associated with each anchor,

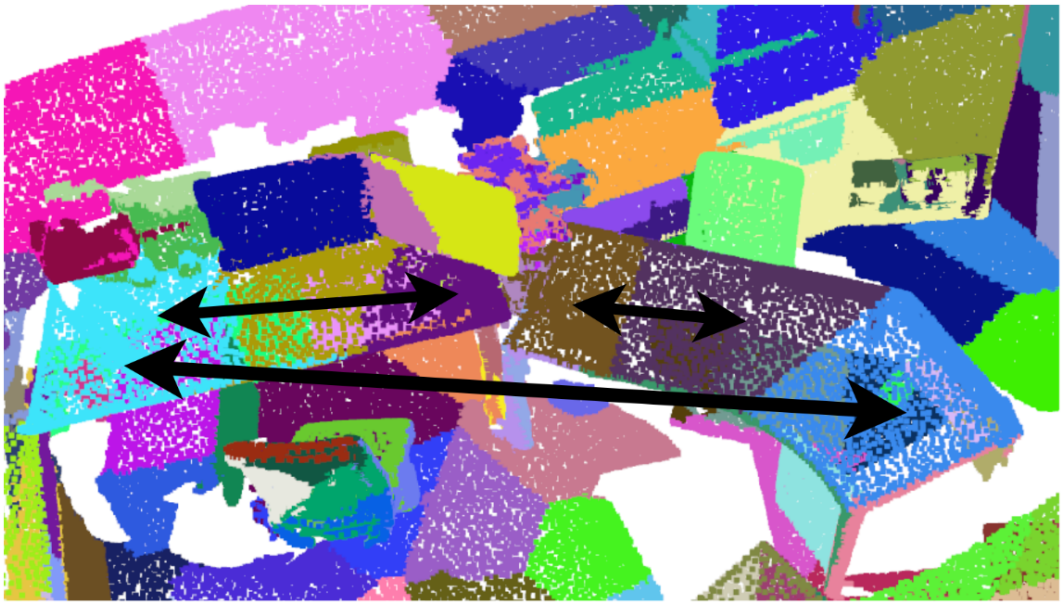


Figure 4.4: Illustration of the sparse label-aware supervised contrastive learning process. The process aims to encourage the clustering of supervoxels belonging to the same class while pushing apart supervoxels from different classes in the latent space.

it ensures that the learning process is fair and not biased towards anchors with a larger number of positive pairs. In section 4.3.1 where we have an unsupervised setting and labeled positive pairs are absent, this normalization factor is not necessary. The incorporation of the normalization factor in Equation 4.5 enables the model to learn discriminative representations effectively by appropriately scaling the loss contributions based on the availability of labeled positive pairs.

The aim of applying supervised contrastive learning is to tune the latent space to better serve our specific requirements. However, we observed that applying supervised contrastive learning to the supervoxels alone is sufficient to construct a meaningful latent space. This is because our unsupervised geometric contrastive learning, which establishes connections between superpixels and supervoxels, indirectly maximizes the similarity between superpixels connected to the same supervoxel. Additionally, considering that generated superpixels can be noisy, we applied the supervised contrastive loss on the supervoxels only. The justification for this choice will be discussed further in the ablation study section 5.5.1.

The supervised contrastive loss facilitates the learning process by encouraging the model to project supervoxels belonging to the same class closer together in the latent space while pushing supervoxels from different classes apart. This helps in distinguishing unlabeled supervoxels that fall closer to a cluster of supervoxels belonging to a specific class. Incorporating the sparse labels into the supervised contrastive learning process enables our model to leverage the limited labeled samples more effectively, enhancing the quality and discriminative power of the learned representations.

By combining unsupervised geometric contrastive learning and sparse label-aware supervised contrastive learning provides a powerful framework for learning rich and discriminative representations from both labeled and unlabeled data. This combination enabled our model to capture both the intrinsic geometric properties of the scene and the semantic information encoded in the sparse labels.

4.4 Pseudo-labeling Techniques

In this section, we introduce our pseudo-labeling techniques, which play a crucial role in leveraging limited labeled samples and enhancing the performance of our segmentation model. Pseudo-labeling allows us to assign labels to unlabeled supervoxels based on certain criteria, expanding the available labeled samples and providing additional supervision during the training process.

Our pseudo-labeling techniques encompass the integration of both 2D and 3D information, thereby leveraging the complementary strengths offered by these modalities. We introduce innovative strategies that incorporate the 2D modality as an additional

verification step within the pseudo-labeling process. This integration contributes to the refinement and accuracy of our pseudo-labeling techniques, elevating their effectiveness in generating reliable labels for the unlabeled supervoxels.

The subsequent sections explore different pseudo-labeling techniques: Pseudo-labeling Leveraging Prediction Probabilities, Pseudo-labeling Leveraging Class Prototypes, and Removal of Incorrect Pseudo-Labels.

4.4.1 Pseudo-labeling Leveraging Prediction Probabilities

This section delineates a methodical pseudo-labeling strategy that incorporates prediction probability thresholds. It involves assigning pseudo-labels to previously unlabeled supervoxels, drawing from the prediction probability scores derived from average-pooled predictions.

Utilizing Average-Pooled Predictions: The process begins by obtaining average-pooled supervoxel predictions from the point cloud. We denote the prediction scores for a given supervoxel as $\mathbf{F}$.

Acquiring Probability Scores with Softmax: The softmax function is applied to the prediction scores $\mathbf{F}$, thus creating a class-based probability distribution. The resulting probability scores, $\mathbf{P} = [p_1, p_2, \ldots, p_C]$, illustrate the likelihood of the supervoxel belonging to class c, where C symbolizes the total class count.

Threshold-based Classification: The pseudo-label for a supervoxel is identified by comparing the highest likelihood probability with a predefined threshold T. If $p_{\max} > T$, where $p_{\max}$ is the highest probability, the supervoxel receives the corresponding class label.

Superpixel Validation: An innovative addition to this pseudo-labeling approach is the inclusion of 2D modality for validation. For each supervoxel, we consider corresponding 2D superpixel prediction scores $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_N$, where N represents the total superpixel count. Each superpixel prediction score $\mathbf{Q}_i$ is compared with the predefined threshold T after applying softmax function. If any of the superpixels satisfy both conditions: (1) the predicted class matches that of the supervoxel, and (2) the associated probability score exceeds threshold T, the pseudo-label decision is validated.

The pseudo-labeling algorithm, as shown in Algorithm 2, outlines the steps involved in the prediction probability-based pseudo-labeling technique with a 2D sanity check. For each supervoxel, the algorithm extracts the prediction scores and computes the probability scores. The class with the highest probability is identified, and if the probability exceeds the threshold, the algorithm proceeds to validate the pseudo-label decision using corresponding 2D superpixel prediction probabilities. If any superpixel satisfies the conditions of matching predicted class and probability exceeding the threshold, the supervoxel is assigned the class label.

Algorithm 2 Pseudo-labeling Leveraging Prediction Probabilities with 2D Sanity Check

Require: Supervoxels $\mathcal{SV}$, 2D Superpixels $\mathcal{SP}$, Threshold T
Ensure: Pseudo-labeled supervoxels

for each supervoxel **sv** in $\mathcal{SV}$ **do**
  Extract the prediction scores **F** for **sv**
  Compute the probability scores $\mathbf{p} = \text{softmax}(\mathbf{F})$
  Identify the class $c = \text{argmax}(\mathbf{p})$ with the highest probability
  if $p_c > T$ **then**
    for each corresponding 2D superpixel **sp** in $\mathcal{SP}[\mathbf{sv}]$ **do**
      Compute the prediction scores **Q** for **sp**
      Calculate the probability scores $\mathbf{q} = \text{softmax}(\mathbf{Q})$
      Determine the class $c' = \text{argmax}(\mathbf{q})$ with the highest probability
      if $q_{c'} > T$ and $c' = c$ **then**
        Assign class c to **sv**
        Break
      end if
    end for
  end if
end for

The proposed pseudo-labeling approach leveraging prediction probabilities offers a reliable and systematic process for assigning labels to previously unlabeled supervoxels, leveraging prediction probabilities. By integrating the 2D modality for verification, we reinforce the robustness of the labeling process, thus significantly enhancing the accuracy of our pseudo-labeling.

4.4.2 Pseudo-labeling Leveraging Class Prototypes

In this section, we present our systematic pseudo-labeling strategy, which utilizes class prototypes stored in a memory bank. Our approach incorporates two distinct metrics, similarity and distance, to determine the threshold for assigning pseudo-labels.

Utilizing Average Pooled Features: Our procedure commences by collecting average pooled features from the prior stage, producing a feature vector of shape (D,) for every supervoxel. The feature vector for a particular supervoxel is designated as **F**.

Constructing a Memory Bank: We formulate a memory bank as a CxD dimensional vector, where C signifies the class count and D corresponds to feature dimensionality. This memory bank, initially populated with random vertical vectors, serves as a store of class prototypes.

Updating the Memory Bank: As the training ensues, we modify the memory bank using a momentum update strategy. The momentum update equation for the memory bank can be represented as:

$$\mathbf{K}_{\bar{c}} \leftarrow m\mathbf{K}\bar{c} + (1-m)\mathbf{F}_j \tag{4.6}$$

Here, $\mathbf{K}_{\bar{c}}$ represents the memory vector for class $\bar{c}$, $\mathbf{F}_j$ stands for the feature vector of sample j in the mini-batch, and m is the momentum coefficient. This momentum update allows the memory bank to accumulate features over time, thus serving as class prototypes.

Similarity-based Pseudo-labeling: We compute the cosine similarity between the feature vector $\mathbf{F}$ of every supervoxel and the class prototypes preserved in the memory bank. The supervoxel receives the class label associated with the class prototype exhibiting the maximum similarity.

$$\text{Similarity}(\mathbf{F}, \mathbf{K}_{\bar{c}}) = \frac{\mathbf{F} \cdot \mathbf{K}\bar{c}}{|\mathbf{F}||\mathbf{K}_{\bar{c}}|} \tag{4.7}$$

Distance-based Pseudo-labeling: We determine the Euclidean distance between the feature vector $\mathbf{F}$ of each supervoxel and the class prototypes within the memory bank. The supervoxel is assigned the class label linked to the class prototype demonstrating the minimum distance.

$$\text{Distance}(\mathbf{F}, \mathbf{K}_{\bar{c}}) = |\mathbf{F} - \mathbf{K}_{\bar{c}}|_2 \tag{4.8}$$

Class-Specific Threshold Calculation: As part of our pseudo-labeling technique, we introduce a more nuanced approach to making labeling decisions by incorporating class-specific statistics. This approach factors in both the mean and the standard deviation of the distances or similarities for each class, which are calculated as we update the memory bank. This class-specific thresholding, as opposed to employing a single threshold value for all classes, renders a more flexible pseudo-labeling process, finely tuned to the distinct statistical attributes of each class, thereby increasing the robustness of our methodology. In the context of similarity-based pseudo-labeling, the threshold is determined as $\mu_{\bar{c}} - \alpha \cdot \sigma_{\bar{c}}$. Conversely, for distance-based pseudo-labeling, the threshold is formulated as $\mu_{\bar{c}} + \alpha \cdot \sigma_{\bar{c}}$. In both instances, $\mu_{\bar{c}}$ and $\sigma_{\bar{c}}$ represent the mean and standard deviation of class $\bar{c}$ for the respective strategy, and α is a hyperparameter.

Superpixel Validation: An innovative addition to our pseudo-labeling approach is the integration of the 2D modality for validation. For each supervoxel, we consider the corresponding 2D superpixel features $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_N$, where N represents the total count of superpixels. Instead of utilizing prediction probabilities as in section 4.4.1, we now compare each superpixel feature $\mathbf{Q}_i$ with the class prototypes stored in the

Algorithm 3 Pseudo-Labeling Leveraging Class Prototypes with 2D Sanity Check

Require: Supervoxels $\mathcal{SV}$, 2D Superpixels $\mathcal{SP}$, Memory Bank K, Threshold T
Ensure: Pseudo-labeled supervoxels

Set `OPERATOR` ▷ Use > for similarity, < for distance
Set `SELECT` ▷ Use `argmax` for similarity, `argmin` for distance
Set `FUNC` ▷ Use `Eq.4.7` for similarity, `Eq.4.8` for distance
for each supervoxel **sv** in $\mathcal{SV}$ **do**
    Calculate the average pooled feature **F** for **sv**
    Compute the scores $\mathbf{S} = \text{FUNC}(\mathbf{F}, K_{\bar{c}})$
    Identify the class $c = \text{SELECT}(\mathbf{S})$
    if S_c `OPERATOR` T **then**
        for each corresponding 2D superpixel **sp** in $\mathcal{SP}[\mathbf{sv}]$ **do**
            Compute the average pooled feature **Q** for **sp**
            Compute the scores $\mathbf{S}_{2D} = \text{FUNC}(\mathbf{Q}, K_{\bar{c}})$
            Determine the class $c' = \text{SELECT}(\mathbf{S}_{2D})$
            if $S_{c'}$ `OPERATOR` T and $c' = c$ **then**
                Assign class c to **sv**
                Break
            end if
        end for
    end if
end for

memory bank. We employ either the similarity equation 4.7 or the distance equation 4.8, along with the class-specific thresholds, to validate the pseudo-label decision. The validation process requires two conditions to be satisfied: (1) the predicted class matches that of the supervoxel, and (2) the computed superpixel threshold is smaller than the class-specific threshold.

The pseudo-labeling algorithm presented in Algorithm 3 outlines the steps involved in our pseudo-labeling technique that leverages class prototypes, incorporating the 2D sanity check. For each supervoxel, the algorithm calculates the average pooled feature **F** and computes the scores using the selected metric (similarity or distance). The class with the highest score is identified, and if the score exceeds the threshold, the algorithm proceeds to validate the pseudo-label decision using the corresponding 2D superpixel features. If any superpixel satisfies the conditions of matching the predicted class and having a superpixel threshold smaller than the class-specific threshold, the supervoxel is assigned the corresponding class label.

The class prototype-based pseudo-labeling technique provides two distinct strategies, namely similarity-based and distance-based, for assigning pseudo-labels to unlabeled supervoxels. By utilizing the class prototypes stored in the memory bank, our approach capitalizes on the geometric proximity of feature vectors to prototypes, assigning pseudo-labels based on maximum similarity or minimum distance. The integration of the 2D modality for validation further enhances the reliability of the pseudo-labeling process.

4.4.3 Removal of Incorrect Pseudo-Labels

To maintain the integrity of the pseudo-labeling process and ensure the consistency between the assigned pseudo-labels and the evolving model predictions, it is necessary to periodically re-evaluate and potentially remove incorrect pseudo-labels. This step helps in filtering out the supervoxels that no longer meet the initial criteria for pseudo-labeling, such as prediction confidence, similarity, or distance.

The removal of incorrect pseudo-labels presented in Algorithm 4 is performed during the later iterations of the training process. After assigning pseudo-labels to the supervoxels based on the initial criteria, the model continues to learn and update its predictions. As the model evolves, there is a possibility that some initially pseudo-labeled supervoxels may no longer satisfy the criteria that were used for their labeling. These supervoxels may become misclassified or exhibit inconsistencies with the updated model predictions.

To address this issue, we incorporate a validation step in which we reassess the pseudo-labeled supervoxels based on the current model predictions. If a pseudo-labeled supervoxel fails to meet the criteria, it is considered an incorrect pseudo-label

and is subsequently removed. The specific criteria for removal may vary depending on the initial pseudo-labeling method used.

Algorithm 4 Removal of Incorrect Pseudo-labels

Require: Pseudo-labeled Supervoxels $\mathcal{SV}$, Model Predictions $\mathcal{P}$
Ensure: Filtered Pseudo-labeled Supervoxels
1: **for** each pseudo-labeled supervoxel **sv** in $\mathcal{SV}$ **do**
2: Retrieve the initial criteria used for pseudo-labeling CRITERIA
3: Retrieve the current model prediction for **sv**
4: **if** CRITERIA(**sv**) no longer satisfies the criteria **then**
5: Remove the pseudo-label from **sv**
6: **end if**
7: **end for**

The removal of incorrect pseudo-labels ensures that the pseudo-labeled data remains consistent with the evolving model predictions. By periodically re-evaluating and removing incorrect pseudo-labels, we ensure the reliability of the pseudo-labeled data.

5 Experiments And Results

In this chapter, we outline the datasets and evaluation metrics that we used for evaluating our method. We present both quantitative and qualitative results to validate the effectiveness of our approach. Furthermore, we carry out ablation studies to analyze the specific contribution of each individual component of our method towards the final results.

5.1 Training Details

In our training process, we employed various strategies to optimize the performance of our proposed approach. The final loss function consisted of different components, including geometric contrastive loss ($L_{\text{cont}_{\text{geo}}}$), supervised contrastive loss ($L_{\text{cont}_{\text{sup}}}$), 2D supervised loss ($L_{\text{sup}_{\text{2D}}}$), and 3D supervised loss ($L_{\text{sup}_{\text{3D}}}$). We employed the cross entropy loss for the sparse labels and the dice loss [66] for the pseudo-labels in both the 2D and 3D supervised losses. The overall loss was computed as a linear combination of these components with specific weights:

$$L = \lambda_{\text{cont}_{\text{geo}}} L_{\text{cont}_{\text{geo}}} + \lambda_{\text{cont}_{\text{sup}}} L_{\text{cont}_{\text{sup}}} + \lambda_{\text{sup}_{\text{2D}}} L_{\text{sup}_{\text{2D}}} + L_{\text{sup}_{\text{3D}}} \tag{5.1}$$

We set the weights $\lambda_{\text{cont}_{\text{geo}}}$ to 0.2, $\lambda_{\text{cont}_{\text{sup}}}$ to 0.1, and $\lambda_{\text{sup}_{\text{2D}}}$ to 0.1 to balance the contributions of each component in the overall loss. The feature size of supervoxels and superpixels was set to 64, providing a suitable dimensionality for effective contrastive learning and pseudo-labeling. In the contrastive learning process, we applied supervised contrastive learning only to supervoxels in addition to the unsupervised learning, setting the temperature parameter τ to 0.4. We also set the prediction confidence threshold for pseudo-labeling at 0.95 and utilized a combined prediction and distance-based approach for pseudo-labeling.

To implement our approach, we utilized the PyTorch [67] framework along with PyTorch Lightning [68] and the Minkowski Engine [9] sparse tensor library. The training process was conducted on two NVIDIA A40 GPUs. We employed the stochastic gradient descent (SGD) [69] optimizer with a base learning rate of 0.01 and a mini-batch size of 16. A polynomial learning rate scheduler with a power of 0.9 was applied to adaptively adjust the learning rate during training. Additionally, we set the momentum and

weight decay parameters to 0.9 and 0.0001, respectively, to stabilize and regularize the training process.

To facilitate faster training, we set the voxel size for the Minkowski Engine [9] to 5 cm, balancing computational efficiency with the level of detail captured in the segmentation. We also resized the 2D images to a smaller resolution of 240 x 320. The model was trained on both the ScanNetv2 [4] and 2D-3D-S [5] dataset for 100 epochs. During initialization, the 2D UNet part of the model was initialized using weights pretrained on ImageNet [70], while the 3D part was initialized from scratch.

5.2 Dataset and Evaluation Metrics

5.2.1 Datasets

We utilized two widely used datasets for evaluating our proposed methods: ScanNetv2 [4] and 2D-3D-S [5].

ScanNetv2 [4]: ScanNetv2 [4] is a large-scale indoor scene dataset containing diverse indoor environments such as offices, living rooms, and other indoor spaces. It consists of over 2.5 million RGB-D frames across 1500 scans, providing rich visual and depth information. The dataset is annotated with 3D camera poses, surface reconstructions, and semantic segmentation labels. ScanNetv2 [4] is officially split into 1201 training scans and 312 validation scans, each captured from different scenes. Additionally, there is a hidden ground truth test set of 100 scans used for benchmarking purposes.

2D-3D-S [5]: 2D-3D-S [5] is a comprehensive dataset that serves as a superset of the S3DIS [6] dataset. It includes the point cloud data from S3DIS [6] along with additional modalities such as RGB and depth images, as well as camera poses and surface normals. 2D-3D-S [5] extends the capabilities of S3DIS [6] by providing richer sensory information for each scene. This dataset enables us to leverage both 2D and 3D modalities in our experiments, enhancing the overall understanding of the scene.

5.2.2 Evaluation Metric

To quantitatively assess the accuracy and quality of our scene segmentation model, we employed the Mean Intersection over Union (mIoU) metric. mIoU is a commonly used evaluation metric in semantic segmentation tasks. It measures the average intersection over union across all object classes in the dataset.

The mIoU is calculated by dividing the sum of the intersection areas between the predicted segmentation masks and the corresponding ground truth masks by the sum of the union areas:

$$\text{mIoU} = \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i + FN_i} \tag{5.2}$$

where C is the total number of object classes, TP_i is the true positive count for class i, FP_i is the false positive count for class i, and FN_i is the false negative count for class i. By using mIoU as our evaluation metric, we can assess the model's ability to accurately segment different object classes in the scene where a higher mIoU score indicates a better segmentation performance.

To quantitatively assess the performance of our pseudo-labeling approach, in addition to mIoU, we also employed precision and recall as evaluation metrics. Precision measures the proportion of correctly labeled positive samples out of the total predicted positive samples, while recall measures the proportion of correctly labeled positive samples out of the total ground truth positive samples.

Precision is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.3}$$

Recall is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.4}$$

where TP represents true positives, FP represents false positives, and FN represents false negatives.

Precision provides insights into the accuracy of the pseudo-labeled samples, indicating how many of the positively labeled samples are actually correct. Recall, on the other hand, provides insights into the completeness of the pseudo-labeling approach, indicating how many of the actual positive samples are correctly identified.

By using precision and recall as additional evaluation metrics, we can assess the effectiveness of our pseudo-labeling method in correctly identifying and labeling the unlabeled samples. These metrics provide a more comprehensive understanding of the performance of our approach and help to evaluate its robustness and reliability in capturing the true positive samples.

5.3 Quantitative Results

On ScanNetv2 [4] we evaluate the performance of our method in four different weakly supervised settings. These settings include scenarios where we used 20 points annotated per scene, as well as 0.01%, 0.02%, and 0.1% of points annotated per scene. The 20

points annotated per scene setting is selected from the official ScanNetv2 [4] data efficient benchmark, while the other settings randomly sample points from each scene. The results of our method on the validation set are presented in Table 5.1, and we also evaluated our method using the official benchmark, as shown in Table 5.2. Notably, our method outperformed SQN [1] in the 0.1% of points per scene setting. While our method achieved competitive results in other settings, we were unable to achieve SOTA performance compared to current weakly supervised methods in those scenarios.

It is important to highlight that SQN [1] reports that PointMatch [3] and OTOC [2] utilize the provided ScanNetv2 [4] segments as supervoxels, assuming that these supervoxels respect object boundaries. However, it is worth noting that this approach may not be applicable to all datasets. For example, 2D-3D-S [5] does not provide segments that can be used as supervoxels. In contrast, our method employs oversegmentation using VCCS [12] to generate supervoxels, which introduces some noise into the process.

For 2D-3D-S [5] we also evaluate the performance in three different scenarios, where 0.01%, 0.02%, and 0.1% of points were annotated per scene. We follow the official train/validation split to train our method on Area 1,2,3,4,6 and report our performance on Area 5. The results of our method on the validation set are presented in Table 5.3. In contrast to our results on the ScanNetv2 [4], our method demonstrated superior performance compared to other SOTA weakly supervised methods in the 0.02% and 0.1% settings.

Method	**Supervision**	**mIoU (%)**
MinkowskiNet [9]	100%	72.2
BPNet [56]	100%	73.9
OTOC [2]	20 points	61.4
OTOC [2]	0.02%	**70.4**
SQN [1]	0.1%	53.5
Pointmatch [3]	20 points	**64.8**
PointMatch [3]	0.01%	**58.7**
Pointmatch [3]	0.1%	**69.3**
Ours	20 points	58.4
Ours	0.01%	54.4
Ours	0.02%	61.1
Ours	0.1%	67.0

Table 5.1: mIoU (%) on different supervision settings on ScanNetv2 validation set.

Method	**Supervision**	**mIoU (%)**
PointNet++ [41]	100%	33.9
SPLATNet [71]	100%	39.3
TangentConv [72]	100%	43.8
PointCNN [73]	100%	45.8
FPConv [74]	100%	63.9
RandLA-Net [45]	100%	64.5
PointConv [47]	100%	66.6
KPConv [48]	100%	68.4
MinkowskiNet [9]	100%	73.6
Virtual MVFusion [55]	100%	74.6
BPNet [56]	100%	74.9
Occuseg [58]	100%	76.4
Mix3D [60]	100%	78.1
OTOC [2]	20 points	59.4
OTOC [2]	0.02%	**69.1**
SQN [1]	0.01%	35.9
SQN [1]	0.1%	56.9
PointMatch [3]	20 points	**62.4**
PointMatch [3]	0.01%	**57.1**
Pointmatch [3]	0.1%	**68.8**
Ours	20 points	54.9
Ours	0.01%	50.1
Ours	0.02%	58.9
Ours	0.1%	66.1

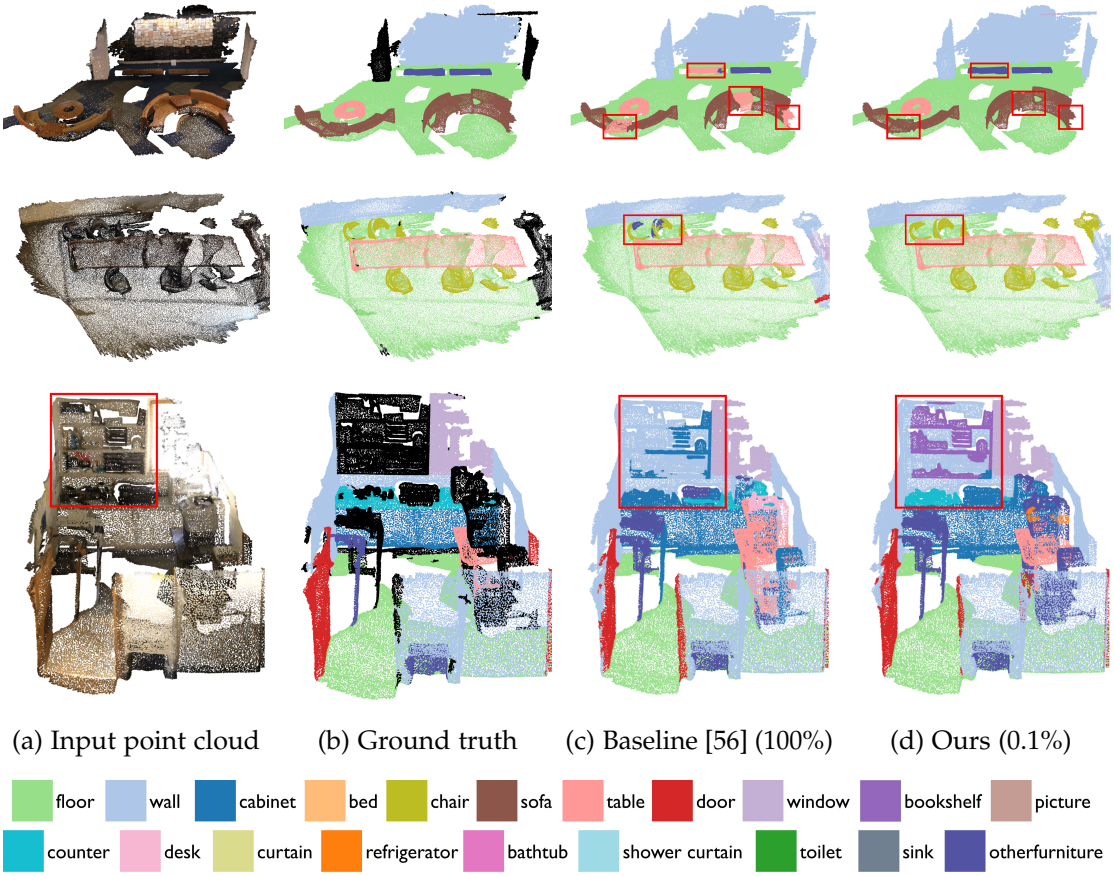
Table 5.2: mIoU (%) on different supervision settings on ScanNetv2 hidden test set.

Method	**Supervision**	**mIoU (%)**
PointNet [40]	100%	41.1
SegCloud [75]	100%	48.9
TangentConv [72]	100%	52.8
PointCNN [73]	100%	57.3
SPGraph [76]	100%	58.0
MinkowskiNet [9]	100%	65.4
Virtual MVFusion [55]	100%	65.4
KPConv [48]	100%	67.1
PointTransformer [77]	100%	70.4
OTOC [2]	0.02%	50.1
SQN [1]	0.01%	45.3
SQN [1]	0.1%	61.4
PointMatch [3]	0.01%	**59.9**
PointMatch [3]	0.1%	63.4
Ours	0.01%	57.2
Ours	0.02%	**61.7**
Ours	0.1%	**63.8**

Table 5.3: mIoU (%) on different supervision settings on 2D-3D-S Area-5.

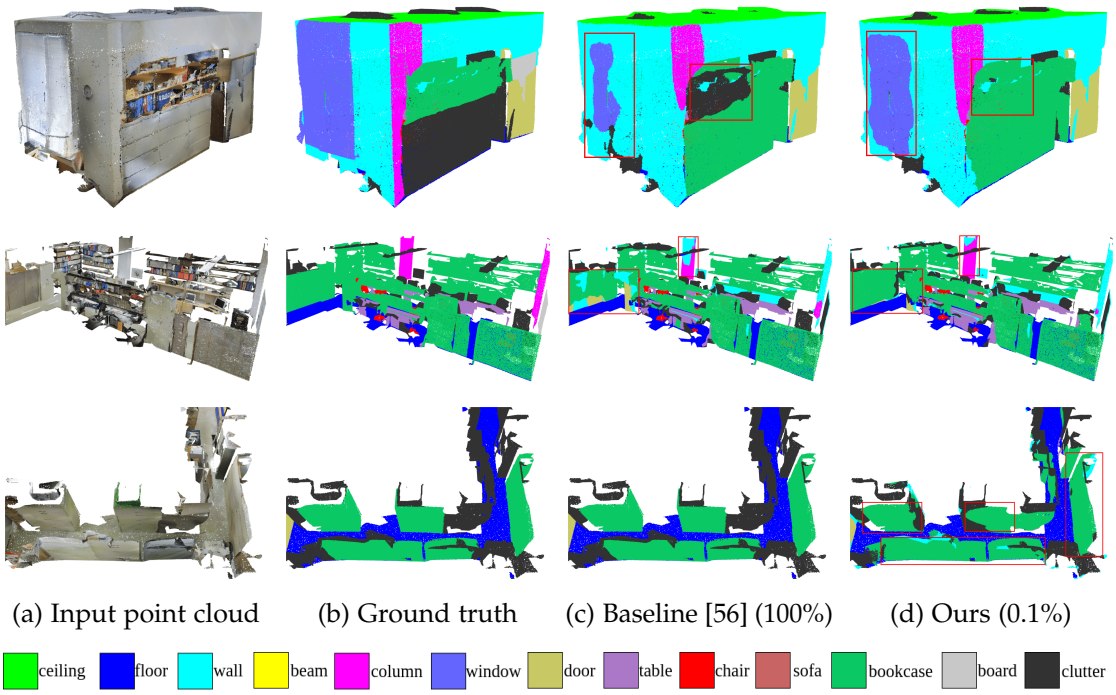
5.4 Qualitative Results

In addition to our quantitative results, we provide qualitative segmentation results to demonstrate the effectiveness of our method. Figure 5.1 showcases the segmentation results on the ScanNetv2 dataset [4], while Figure 5.2 presents the results on the 2D-3D-S dataset [5]. Each figure includes the following visualizations: (a) the colored input point cloud, (b) the ground truth (GT) semantic segmentation, (c) the semantic predictions of our baseline [56] model trained with 100% of labels, and (d) the predictions of our method trained with 0.1% of labels.



(a) Input point cloud (b) Ground truth (c) Baseline [56] (100%) (d) Ours (0.1%)

Figure 5.1: Qualitative segmentation results on the ScanNetv2 dataset. (a) Colored input point cloud. (b) Ground truth semantic segmentation. (c) Semantic predictions of our baseline model trained with 100% of labels. (d) Semantic predictions of our method trained with 0.1% of labels.



(a) Input point cloud (b) Ground truth (c) Baseline [56] (100%) (d) Ours (0.1%)

Figure 5.2: Qualitative segmentation results on the 2D-3D-S dataset. (a) Colored input point cloud. (b) Ground truth semantic segmentation. (c) Semantic predictions of our baseline model trained with 100% of labels. (d) Semantic predictions of our method trained with 0.1% of labels.

5.5 Ablation Study

In this section, we conduct an ablation study to evaluate the effectiveness of various components and techniques used in our approach. We aim to understand the impact of each component on the segmentation performance and gain insights into their contributions. The ablation study allows us to analyze the influence of individual design choices and methodologies, providing a deeper understanding of the key factors driving the results. The ablation studies are conducted under the 20 annotated labels per scene setting.

To facilitate clarity and consistency, we use specific names to refer to key components: Geometric contrastive loss (from Section 4.3.1), supervised contrastive loss (from Section 4.3.2), prediction-based pseudo-labeling (from Section 4.4.1), similarity-based pseudo-labeling (from Section 4.4.2 with a similarity-based approach), and distance-based pseudo-labeling (from Section 4.4.2 with a distance-based approach).

5.5.1 Importance of Contrastive Learning

The table 5.4 presents the results of our ablation study, which investigates the impact of contrastive learning in the scene segmentation model. By analyzing the results, we can gain insights into the effectiveness of different contrastive learning components.

Method	**mIoU (%)**
Baseline BPNet [56]	55.92
+ Geometric contrastive loss	56.75
+ 3D Supervised contrastive loss	**57.39**
+ 2D Supervised contrastive loss	56.03

Table 5.4: Effect of different supervised/self-supervised contrastive losses on mIoU (%). Each line starting with "+" represents the incremental addition of the specific loss to the previous row, building upon the previous model configuration.

One of the key observations is that the inclusion of the geometric contrastive loss significantly improves the performance of semantic segmentation. This loss enables the model to capture geometric relationships and spatial information, resulting in more accurate segmentations. This finding supports the importance of leveraging geometric cues in scene understanding tasks.

Furthermore, when incorporating the supervised contrastive loss with the 3D modalities, we observe a notable enhancement in the segmentation results. This demonstrates

that leveraging the discriminative features learned from 3D supervoxels leads to improved semantic segmentation. The supervised contrastive loss encourages the model to differentiate between supervoxels belonging to different classes while promoting similarity among those belonging to the same class.

However, the addition of the supervised contrastive loss with the 2D modality does not show a substantial improvement in the segmentation performance. This can be attributed to the fact that the geometric loss, combined with the supervised contrastive loss in the 3D space, already effectively contrasts the superpixels belonging to the same supervoxel, thus capturing the intra-class similarity. As a result, the inclusion of the supervised contrastive loss with the 2D modality does not yield significant additional benefits.

These findings highlight the importance of incorporating contrastive learning techniques, such as geometric and supervised contrastive losses, in improving the scene segmentation model. Leveraging geometric cues and exploiting discriminative features learned from the 3D modalities can significantly enhance the model's segmentation performance, while the additional use of the 2D modality may not offer substantial improvements due to redundancy in the learned representations.

5.5.2 Temperature in contrastive learning

In this subsection, we investigate the impact of temperature on the performance of contrastive learning. The temperature parameter is a crucial component in contrastive learning algorithms, as it controls the concentration of the probability distribution over negative samples during the contrastive loss computation.

The results of our ablation study, presented in Table 5.5, shows the performance of the model with different temperature values..

Temperature	**mIoU (%)**
0.07	54.64
0.1	56.03
0.2	55.22
0.4	**57.04**
0.6	56.76

Table 5.5: Effect of different temperature values on segmentation performance, measured by mIoU (%).

Analyzing the results, we can observe that the choice of temperature has a noticeable impact on the segmentation performance. Lower temperature values (0.07 and 0.1) lead

to lower mIoU scores, indicating a less effective learning of discriminative features. As the temperature increases to 0.2, we see a slight improvement in the mIoU score. However, the best performance is achieved at a temperature of 0.4, where we observe the highest mIoU of 57.04%. Further increasing the temperature to 0.6 leads to a slight decrease in the mIoU score.

These results suggest that the selection of an appropriate temperature value is crucial in contrastive learning. A higher temperature allows for a more diverse exploration of negative samples, enabling the model to better differentiate between similar and dissimilar instances. However, a high temperature value can lead to excessive diversity and weaken the discriminative power of the learned representations.

Based on our ablation study, we select a temperature value of 0.4 for our contrastive learning experiments, as it consistently yields the best segmentation performance. This optimal temperature strikes a balance between effective contrastive learning and discriminative feature extraction in the scene segmentation model.

5.5.3 Different pseudo-labeling approaches

We conduct a comprehensive analysis of different pseudo-labeling approaches and their impact on the segmentation results. Table 5.6 presents the evaluation results for prediction, similarity, and distance-based pseudo-labeling strategies. Notably, the prediction approach achieves the highest 3D mIoU score, while the distance-based approach demonstrates superior precision and recall.

Method	**Modality**	**mIoU (%)**	**Precision**	**Recall**	**Pseudo-labeled Supervoxels**
Prediction	3D	**59.08**	65.28	74.26	1.959.697
Prediction	2D + 3D	58.71	70.44	72.60	1.376.372
Similarity	3D	57.14	52.73	73.14	1.959.736
Similarity	2D + 3D	56.96	45.95	71.69	1.622.384
Distance	3D	57.31	55.24	73.33	1.959.736
Distance	2D + 3D	57.28	**85.45**	**91.41**	17.847

Table 5.6: Comparison of different pseudo-labeling approaches and their impact on the segmentation results in terms of mIoU (%). The prediction approach achieves the highest 3D mIoU score, while the distance-based approach demonstrates superior precision and recall.

When comparing the 3D and 2D + 3D approaches, we observe a decline in the

total number of pseudo-labeled supervoxels upon incorporating the 2D modality. This decline can be attributed to the alignment challenges between pseudo-labeling checks and the corresponding superpixels. Consequently, fewer supervoxels meet the criteria for confident pseudo-labeling. However, despite the inclusion of the 2D modality, we do not observe a significant improvement in the 3D and 2D mIoU scores. This finding prompts a deeper investigation into the underlying factors impeding the performance enhancement.

Furthermore, the addition of the 2D modality does not consistently enhance precision and recall across all pseudo-labeling approaches. In the prediction-based approach, the recall exhibits a decrease when incorporating the 2D modality. Similarly, both precision and recall decrease in the similarity-based approach. In contrast, the distance-based approach demonstrates improved precision and recall with the inclusion of the 2D check. Surprisingly, despite the gains in precision and recall, the mIoU decreases for the distance-based approach. Moreover, the total number of pseudo-labeled supervoxels experiences a substantial reduction, with only approximately 17000 supervoxels being pseudo-labeled.

To elucidate the superior mIoU performance of the prediction-based approach compared to other strategies, a more in-depth analysis is required. The prediction-based approach likely harnesses the inherent semantic information present in the input data, enabling effective discrimination between supervoxels belonging to different classes. This capability to capture fine-grained class boundaries and semantic details contributes to the higher mIoU scores achieved.

5.5.4 Combining Prediction and Distance-Based Pseudo-Labeling

We also investigated the impact of incorporating prediction confidence as an additional check to the distance-based pseudo-labeling approach. By introducing prediction confidence as a criterion for pseudo-labeling, we aim to improve the precision and recall of the generated pseudo-labeled samples.

We observe that the prediction approach achieves the highest 3D mIoU score, indicating its effectiveness in capturing fine-grained class boundaries and semantic details. On the other hand, the distance-based approach demonstrates superior precision and recall, emphasizing its ability to accurately identify supervoxels belonging to different classes.

Motivated by the strengths exhibited by both approaches, we decide to combine the prediction and distance-based methods to leverage their complementary advantages. Table 5.7 presents the results obtained with distance-based approach and combined approach. This combined approach capitalizes on the predictive power of the prediction approach to capture detailed semantic information, while also benefiting from the

precision and recall improvements provided by the distance-based approach.

Method	mIoU (%)	Precision	Recall	Pseudo-labeled Supervoxels
Distance	57.28	85.45	91.41	17.847
Combined	**58.21**	**89.45**	**93.73**	14.333

Table 5.7: Combining Prediction and Distance-Based Approaches. The combined approach shows improved mIoU (%), precision, and recall compared to the distance-based approach alone.

The decision to incorporate prediction confidence as an additional check aims to enhance the reliability and quality of the pseudo-labeling process. By considering the confidence level of predictions, we filter out uncertain or noisy pseudo-labels, resulting in more reliable and accurate segmentation results.

The experimental results support the effectiveness of the combined approach. By combining the prediction and distance-based methods, we achieve a higher 3D mIoU score compared to the distance-based approach alone. Furthermore, the precision and recall metrics also show improvements, indicating a more precise and comprehensive classification of supervoxels.

The successful integration of prediction confidence as an additional check highlights the potential of combining different approaches to enhance the segmentation performance.

5.5.5 Different Thresholds for Combined Prediction and Distance-Based Pseudo-Labeling

In this section, we explore the impact of employing different distance thresholds in the combined distance and prediction-based pseudo-labeling approach in section 5.5.4.

As discussed in Section 4.4.2, distance-based pseudo-labeling involves class-specific threshold calculations using the mean ($\mu_{\bar{c}}$) and standard deviation ($\sigma_{\bar{c}}$) values for each class. Distance-based pseudo-labeling uses $\mu_{\bar{c}} + \alpha \cdot \sigma_{\bar{c}}$ for class $\bar{c}$, where assigning smaller values to α denotes stricter thresholding.

By varying the threshold values, we aim to gain insights into their influence on the segmentation performance and understand the trade-offs between accuracy and completeness.

The results of our experiments, as shown in Table 5.8, reveal interesting observations regarding the effect of different thresholds on the segmentation results. We find that

stricter thresholds lead to improvements in precision and recall metrics, indicating a higher degree of accuracy in differentiating supervoxels belonging to distinct classes. However, it is noteworthy that as the threshold becomes stricter, there is a corresponding decrease in the 3D mIoU score, which serves as a measure of overall segmentation performance.

Threshold	mIoU (%)	Precision	Recall	Pseudo-labeled Supervoxels
α=-1	58.21	**89.45**	**93.73**	14.333
α=0	58.61	84.22	92.74	499.445
α=1	59.05	78.87	85.91	957.650
α=2	**59.06**	73.03	82.85	1.152.212

Table 5.8: Segmentation performance mIoU(%) with different thresholds for the combined approach. The results show that stricter thresholds improve precision and recall but lead to a decrease in 3D mIoU (%).

The decrease in 3D mIoU suggests that while stricter thresholds enhance the precision and recall of pseudo-labeled samples, they may also result in the exclusion of supervoxels that could contribute to a more comprehensive and holistic segmentation. This trade-off between accuracy and completeness highlights the importance of selecting an appropriate distance threshold that balances the objectives of precise class labeling and capturing the full extent of the scene's semantic information.

Furthermore, we observe that the total number of pseudo-labeled supervoxels decreases as the threshold becomes stricter. This reduction indicates a higher level of selectivity in the pseudo-labeling process, with a more focused and refined set of labeled supervoxels being retained.

The findings from our experiments emphasize the need for a thoughtful consideration of the choice of distance threshold in the combined pseudo-labeling approach. It is crucial to strike a balance between precision, recall, and 3D mIoU score, taking into account the specific requirements and objectives of the segmentation task at hand.

5.5.6 Relaxation of 2D Pseudo-Labeling Criteria

We explored the impact of relaxing the criteria for 2D pseudo-labeling by disabling the threshold checking in the 2D sanity part. Instead, the only requirement for pseudo-labeling was having the same class assignment between the 2D and 3D modalities. The objective was to incorporate a broader range of information and potentially increase the coverage of pseudo-labeled samples.

Table 5.9 presents the results of the relaxed distance-based pseudo-labeling approach compared to the non-relaxed approach. Surprisingly, the non-relaxed approach achieved better performance, with a higher 3D mIoU, precision, and recall compared to the relaxed approach. The non-relaxed approach yielded a 3D mIoU of 59.06, while the relaxed approach achieved a slightly lower 3D mIoU of 58.85. This indicates that the stricter criteria for 2D pseudo-labeling led to improved segmentation accuracy.

The results suggest that maintaining the threshold in the 2D sanity part, which ensures a closer alignment between the 2D and 3D modalities, is beneficial for the segmentation performance.

Method	**mIoU (%)**	**Precision**	**Recall**	**Pseudo-labeled Supervoxels**
Non-relaxed	59.06	73.03	82.85	1.152.212
Relaxed	58.85	65.98	81.97	1.563.619

Table 5.9: Impact of relaxing 2D pseudo-labeling criteria on segmentation performance, measured by mIoU (%). The non-relaxed approach outperforms the relaxed approach, indicating the importance of maintaining stricter criteria.

5.5.7 Relaxation of Threshold during Training

We explore the idea of using a dynamic threshold during the training process instead of a fixed threshold for pseudo-labeling. The motivation behind this approach is to leverage the benefits of a stricter threshold in the early stages of training, where precise and accurate pseudo-labels can be advantageous for contrastive learning and class prototype assignment. As the training progresses, we gradually relax the threshold to allow a broader range of supervoxels to be pseudo-labeled, aiming to enhance the semantic segmentation performance.

However, our experimental results, as shown in Table 5.10, reveal that using a constant threshold throughout the training process yields better results in terms of 3D mIoU. The constant threshold maintains a consistent criterion for pseudo-labeling, ensuring a more stable and reliable training process. Although the dynamic threshold approach demonstrates higher precision and recall in the pseudo-labeling process, it does not translate into improved 3D mIoU scores.

These findings suggest that the stability and consistency provided by a constant threshold are crucial for optimizing the semantic segmentation performance. While a dynamic threshold may enable a broader coverage of supervoxels and potentially

Method	mIoU (%)	Precision	Recall	Pseudo-labeled Supervoxels
Constant Threshold	59.06	73.03	82.85	1.152.212
Dynamic Threshold	58.50	75.53	87.07	765.510

Table 5.10: Impact of threshold relaxation during training on segmentation performance, measured by mIoU (%). Despite higher precision and recall with a dynamic threshold, a constant threshold yields better mIoU, emphasizing the importance of stability and consistency in pseudo-labeling.

capture a wider range of semantic classes, it also introduces more noise and may compromise the accuracy of the pseudo-labeling process.

5.5.8 Removal of wrongly classified pseudo-labels

We implemented a mechanism to periodically remove wrongly classified pseudo-labels to ensure the integrity of the pseudo-labeling process. Table 5.11 presents the results obtained with and without the removal of wrongly classified pseudo-labels. Our evaluation showed that the removal of wrongly classified pseudo-labels had a limited impact on the mIoU and recall, but we observed a slight improvement in precision. The total number of pseudo-labels remained relatively consistent, suggesting that the model maintained its confidence in the assigned labels.

Method	mIoU (%)	Precision	Recall	Pseudo-labeled Supervoxels
Without removal	59.06	73.03	82.85	1.152.212
With removal	58.99	73.53	82.83	1.151.730

Table 5.11: Impact of removing wrongly classified pseudo-labels on segmentation performance, measured by mIoU (%). While the removal process has a limited impact on mIoU and recall, it slightly improves precision, indicating a more accurate pseudo-labeling process.

However, our approach of using the same model for both pseudo-labeling and segmentation might have limitations in accurately identifying and discarding wrongly classified pseudo-labels which may be a reason why OTOC [2] employed a separate model for pseudo-labeling.

5.5.9 Different Oversegmentation Methodologies

ScanNetv2 [4] provides a conservative oversegmentation of the mesh through a graph cut method, resulting in a clean oversegmentation that preserves object boundaries without introducing noise. We refer to this oversegmentation as GT oversegmentation. While propagating sparse labels to the oversegmented supervoxels using the GT oversegmentation, the learning process remains free from noise, and the average pooled features learned from these supervoxels better capture the characteristics of each class, as they align with the object boundaries.

Table 5.12 presents the quantitative comparison between using the oversegmentation acquired from VCCS and GT oversegmentation for our methodology.

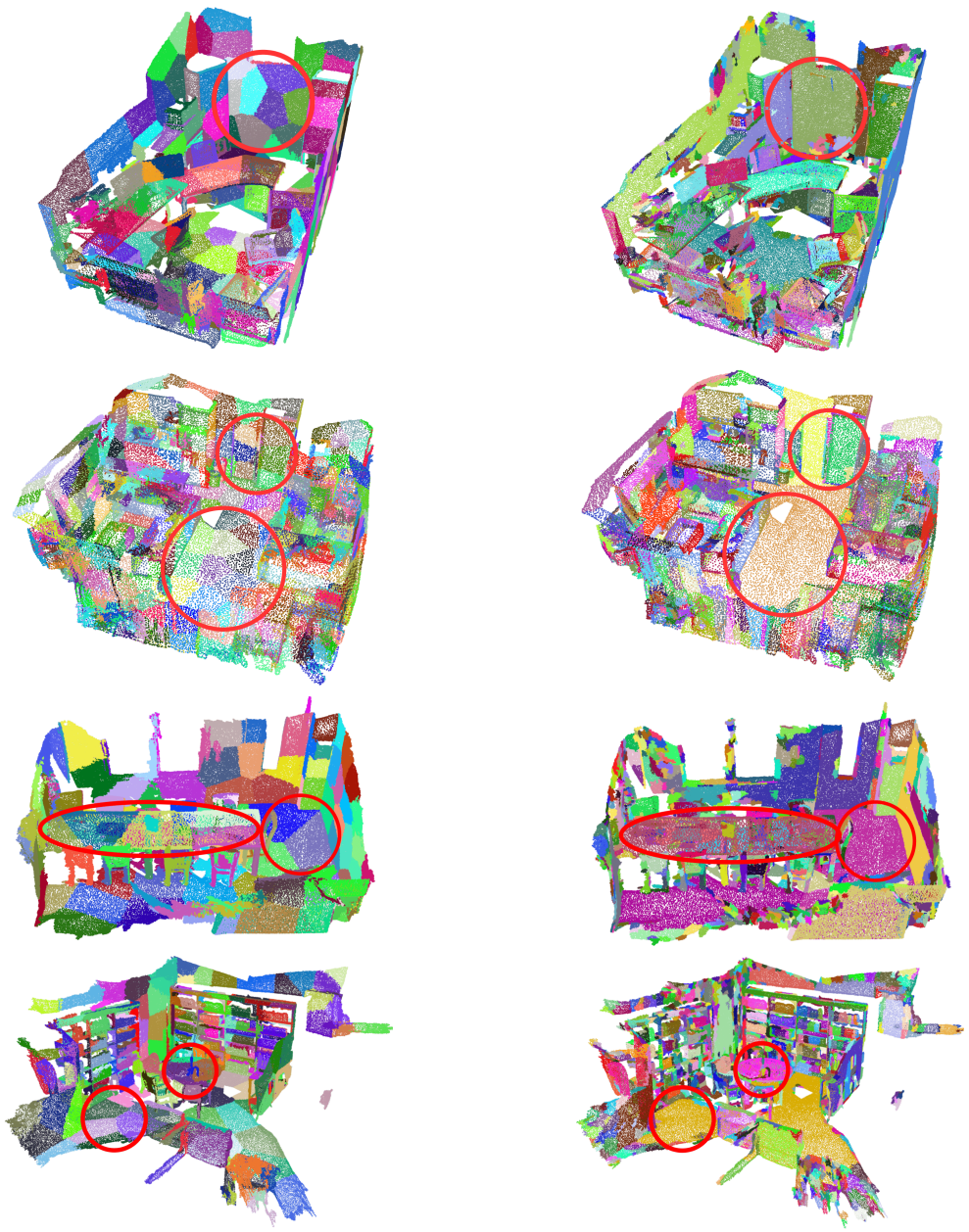
Oversegmentation	**mIoU (%)**	**Precision**	**Recall**	**Pseudo-labeled Supervoxels**
VCCS [12]	59.06	73.03	82.85	1.152.212
GT	59.36	73.86	85.33	440.843

Table 5.12: Comparative performance of VCCS and GT oversegmentation methods in terms of mIoU (%). Despite a lower number of pseudo-labeled supervoxels, GT oversegmentation yields superior segmentation results, highlighting the importance of accurate initial oversegmentation.

Our comparative analysis between VCCS [12] oversegmentation and GT oversegmentation, as utilized in OTOC [2], reveals the superior performance achieved with GT oversegmentation in terms of 3D mIoU, precision, and recall. Despite the significantly lower number of pseudo-labeled supervoxels obtained through GT oversegmentation, it yields better segmentation results.

In our exploration of different pseudo-labeling approaches in section 5.5.3, we established that a higher number of pseudo-labeled supervoxels generally leads to improved segmentation scores. However, in the case of GT oversegmentation, we observed that even with a lower number of pseudo-labeled supervoxels, the segmentation quality is enhanced. This phenomenon can be attributed to the cleaner and less noisy nature of GT oversegmentation, enabling the selection of a smaller yet more reliable set of supervoxels for pseudo-labeling. This finding underscores the significant impact of oversegmentation noise on the overall segmentation performance.

While the utilization of GT oversegmentation proves advantageous for semantic segmentation, it is important to acknowledge that obtaining ground truth oversegmentation in practical scenarios is infeasible. Nevertheless, this ablation study underscores the critical role of accurate initial oversegmentation in the pseudo-labeling of point



(a) VCCS [12] (b) GT

Figure 5.3: Visual comparison of VCCS and GT oversegmentation methods. The images illustrate the differences in oversegmentation quality between the two methods, with GT oversegmentation providing cleaner and less noisy supervoxel boundaries.

clouds, reaffirming the importance of addressing oversegmentation noise for optimal segmentation outcomes. The visual comparison of VCCS and GT oversegmentation methods, as shown in Figure 5.3, further illustrates the differences in oversegmentation quality between the two methods, with GT oversegmentation providing cleaner and less noisy oversegmentations.

6 Conclusion

The work presented in this thesis aimed to tackle the challenges associated with weakly supervised point cloud semantic segmentation, introducing an innovative approach that leverages multi-modal information and advanced pseudo-labeling techniques. Our methodology effectively brings together the valuable geometric information from 2D and 3D correspondences, thereby enhancing the results of point cloud segmentation.

One of the cornerstone contributions of this research is the development of a comprehensive framework that integrates multi-modal information into the contrastive learning process. This robust framework leverages oversegmentation to tackle sparse labels, enhancing class representations and leading to a more informative latent space construction. As a result, the overall performance of the contrastive framework is enhanced.

Another significant achievement of this work is the incorporation of the 2D modality into the pseudo-labeling process. This approach allowed us to take advantage of the complementary information provided by the 2D modality, generating accurate and reliable confidence pseudo-labels that guide the learning process and boost segmentation performance.

Furthermore, our methodology introduces an online adaptive pseudo-labeling mechanism, which dynamically adapts to evolving model predictions, eliminating the need for an additional network for generating pseudo-labels. This makes the process more efficient and scalable.

Despite these significant contributions, the research also revealed areas that require further investigation. Notably, the integration of the 2D modality into the pseudo-labeling process improved the precision and recall of pseudo-labeling, but it did not lead to a corresponding increase in the mIoU compared to pseudo-labeling using only the 3D modality. This unexpected outcome suggests potential for further refinement in our methodology, such as addressing the noise in oversegmentation, and superpixel generation.

The exploration of this noise in superpixel generation will be a focus of future research. By understanding and mitigating this noise, it could be possible to more effectively leverage the 2D modality, leading to improvements in the mIoU score. This will open up new opportunities for refining the current methodology and developing more effective semantic segmentation models.

Abbreviations

CRF conditional random field

FCN Fully Convolutional Network

ViT Vision Transformer

CNN Convolutional Neural Network

BPNet Bidirectional Projection Network

BPM bidirectional projection module

SSCN Submanifold Sparse Convolutional Networks

MCNN Minkowski Convolutional Neural Networks

SLIC Simple Linear Iterative Clustering

SEEDS Superpixels Extracted via Energy-Driven Sampling

VCCS Voxel Cloud Connectivity Segmentation

SupContrast Supervised Contrastive Learning

SupCon Supervised Contrastive Loss

OTOC One Thing One Click

SQN Semantic Query Network

FCGF The Fully Convolutional Geometric Features

SLidR Superpixel-driven Lidar Representations

SLIC Simple Linear Iterative Clustering

SGD stochastic gradient descent

mIoU Mean Intersection over Union

GT ground truth

SOTA state of the art

DPT Dense Prediction Transformer

SEEDS Superpixels Extracted via Energy-Driven Sampling

List of Figures

List of Tables

Bibliography

[1] Q. Hu, B. Yang, G. Fang, Y. Guo, A. Leonardis, N. Trigoni, and A. Markham. "Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds." In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*. Springer. 2022, pp. 600–619.

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

[3] Y. Wu, Z. Yan, S. Cai, G. Li, Y. Yu, X. Han, and S. Cui. "Pointmatch: a consistency training framework for weakly supervisedsemantic segmentation of 3d point clouds." In: *arXiv preprint arXiv:2202.10705* (2022).

[4] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. "Scannet: Richly-annotated 3d reconstructions of indoor scenes." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5828–5839.

[5] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. "Joint 2D-3D-Semantic Data for Indoor Scene Understanding." In: *ArXiv e-prints* (Feb. 2017). arXiv: 1702.01105 [cs.CV].

[6] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. "3D Semantic Parsing of Large-Scale Indoor Spaces." In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. 2016.

[7] R. B. Rusu and S. Cousins. "3D is here: Point Cloud Library (PCL)." In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 1–4. DOI: 10.1109/ICRA.2011.5980567.

[8] B. Graham, M. Engelcke, and L. van der Maaten. "3D Semantic Segmentation With Submanifold Sparse Convolutional Networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[9] C. Choy, J. Gwak, and S. Savarese. "4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[10] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. "SLIC superpixels compared to state-of-the-art superpixel methods." In: *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012), pp. 2274–2282.

[11] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool. "SEEDS: Superpixels extracted via energy-driven sampling." In: *ECCV (7)* 7578 (2012), pp. 13–26.

[12] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter. "Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2013.

[13] L. Landrieu and M. Boussaha. "Point cloud oversegmentation with graph-structured deep metric learning." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7440–7449.

[14] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. "A survey on contrastive self-supervised learning." In: *Technologies* 9.1 (2020), p. 2.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. "A Simple Framework for Contrastive Learning of Visual Representations." In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 1597–1607.

[16] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton. "Big self-supervised models are strong semi-supervised learners." In: *Advances in neural information processing systems* 33 (2020), pp. 22243–22255.

[17] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. "Momentum contrast for unsupervised visual representation learning." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

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

[19] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. "Supervised contrastive learning." In: *Advances in neural information processing systems* 33 (2020), pp. 18661–18673.

[20] J. Shotton, J. Winn, C. Rother, and A. Criminisi. "TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context." In: *Int. Journal of Computer Vision (IJCV)* (Jan. 2009).

[21] X. Ren, C. C. Fowlkes, and J. Malik. "Figure/Ground Assignment in Natural Images." In: *Computer Vision – ECCV 2006*. Ed. by A. Leonardis, H. Bischof, and A. Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 614–627. ISBN: 978-3-540-33835-2.

[22] X. He, R. Zemel, and M. Carreira-Perpinan. "Multiscale conditional random fields for image labeling." In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 2. 2004, pp. II–II. DOI: 10.1109/CVPR.2004.1315232.

[23] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[24] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation." In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[25] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu. "ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 162 (2020), pp. 94–114.

[26] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[27] H. Noh, S. Hong, and B. Han. "Learning deconvolution network for semantic segmentation." In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1520–1528.

[28] V. Badrinarayanan, A. Kendall, and R. Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[29] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.

[30] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. "Rethinking atrous convolution for semantic image segmentation." In: *arXiv preprint arXiv:1706.05587* (2017).

[31] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation." In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 801–818.

[32] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. "Pyramid scene parsing network." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2881–2890.

[33] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia. "Psanet: Point-wise spatial attention network for scene parsing." In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 267–283.

[34] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. "Dual attention network for scene segmentation." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3146–3154.

[35] R. Ranftl, A. Bochkovskiy, and V. Koltun. "Vision transformers for dense prediction." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12179–12188.

[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." In: *arXiv preprint arXiv:2010.11929* (2020).

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

[38] G. Riegler, A. Osman Ulusoy, and A. Geiger. "Octnet: Learning deep 3d representations at high resolutions." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3577–3586.

[39] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. "O-cnn: Octree-based convolutional neural networks for 3d shape analysis." In: *ACM Transactions On Graphics (TOG)* 36.4 (2017), pp. 1–11.

[40] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. "Pointnet: Deep learning on point sets for 3d classification and segmentation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.

[41] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." In: *Advances in neural information processing systems* 30 (2017).

[42] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. "Dynamic graph cnn for learning on point clouds." In: *Acm Transactions On Graphics (tog)* 38.5 (2019), pp. 1–12.

[43] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao. "SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters." In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[44] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia. "Pointweb: Enhancing local neighborhood features for point cloud processing." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5565–5573.

[45] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham. "RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[46] S. Wang, S. Suo, W.-C. Ma, A. Pokrovsky, and R. Urtasun. "Deep parametric continuous convolutional neural networks." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2589–2597.

[47] W. Wu, Z. Qi, and L. Fuxin. "Pointconv: Deep convolutional networks on 3d point clouds." In: *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 2019, pp. 9621–9630.

[48] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas. "KPConv: Flexible and Deformable Convolution for Point Clouds." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[49] J. Schult, F. Engelmann, T. Kontogianni, and B. Leibe. "DualConvMesh-Net: Joint Geodesic and Euclidean Convolutions on 3D Meshes." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[50] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan. "Graph Attention Convolution for Point Cloud Semantic Segmentation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[51] L. Jiang, H. Zhao, S. Liu, X. Shen, C.-W. Fu, and J. Jia. "Hierarchical Point-Edge Interaction Network for Point Cloud Semantic Segmentation." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[52] G. Li, M. Muller, A. Thabet, and B. Ghanem. "DeepGCNs: Can GCNs Go As Deep As CNNs?" In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[53] A. Dai and M. Niessner. "3DMV: Joint 3D-Multi-View Prediction for 3D Semantic Scene Segmentation." In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[54] M. Jaritz, J. Gu, and H. Su. "Multi-View PointNet for 3D Scene Understanding." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2019.

[55] A. Kundu, X. Yin, A. Fathi, D. Ross, B. Brewington, T. Funkhouser, and C. Pantofaru. "Virtual Multi-view Fusion for 3D Semantic Segmentation." In: *Computer Vision – ECCV 2020*. Ed. by A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm. Cham: Springer International Publishing, 2020, pp. 518–535. ISBN: 978-3-030-58586-0.

[56] W. Hu, H. Zhao, L. Jiang, J. Jia, and T.-T. Wong. "Bidirectional Projection Network for Cross Dimension Scene Understanding." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 14373–14382.

[57] Z. Hu, M. Zhen, X. Bai, H. Fu, and C.-l. Tai. "JSENet: Joint Semantic Segmentation and Edge Detection Network for 3D Point Clouds." In: *Computer Vision – ECCV 2020*. Ed. by A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm. Cham: Springer International Publishing, 2020, pp. 222–239. ISBN: 978-3-030-58565-5.

[58] L. Han, T. Zheng, L. Xu, and L. Fang. "OccuSeg: Occupancy-Aware 3D Instance Segmentation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[59] Y.-Q. Yang, Y.-X. Guo, J.-Y. Xiong, Y. Liu, H. Pan, P.-S. Wang, X. Tong, and B. Guo. *Swin3D: A Pretrained Transformer Backbone for 3D Indoor Scene Understanding*. 2023. arXiv: 2304.06906 [cs.CV].

[60] A. Nekrasov, J. Schult, O. Litany, B. Leibe, and F. Engelmann. "Mix3D: Out-of-Context Data Augmentation for 3D Scenes." In: *International Conference on 3D Vision (3DV)*. 2021.

[61] C. Choy, J. Park, and V. Koltun. "Fully convolutional geometric features." In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 8958–8966.

[62] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany. "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding." In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer. 2020, pp. 574–591.

[63] A. v. d. Oord, Y. Li, and O. Vinyals. "Representation learning with contrastive predictive coding." In: *arXiv preprint arXiv:1807.03748* (2018).

[64] C. Sautier, G. Puy, S. Gidaris, A. Boulch, A. Bursuc, and R. Marlet. "Image-to-lidar self-supervised distillation for autonomous driving data." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9891–9901.

[65] J. Hou, S. Xie, B. Graham, A. Dai, and M. Nießner. "Pri3d: Can 3d priors help 2d representation learning?" In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5693–5702.

[66] F. Milletari, N. Navab, and S.-A. Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation." In: *2016 fourth international conference on 3D vision (3DV)*. Ieee. 2016, pp. 565–571.

[67] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. "Pytorch: An imperative style, high-performance deep learning library." In: *Advances in neural information processing systems* 32 (2019).

[68] W. Falcon and The PyTorch Lightning team. *PyTorch Lightning*. Version 1.4. Mar. 2019. DOI: 10.5281/zenodo.3828935.

[69] L. Bottou. "Large-scale machine learning with stochastic gradient descent." In: *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*. Springer. 2010, pp. 177–186.

[70] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database." In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[71] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz. "Splatnet: Sparse lattice networks for point cloud processing." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2530–2539.

[72] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou. "Tangent convolutions for dense prediction in 3d." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3887–3896.

[73] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. "Pointcnn: Convolution on x-transformed points." In: *Advances in neural information processing systems* 31 (2018).

[74] Y. Lin, Z. Yan, H. Huang, D. Du, L. Liu, S. Cui, and X. Han. "Fpconv: Learning local flattening for point convolution." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 4293–4302.

[75] 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)*. IEEE. 2017, pp. 537–547.

[76] L. Landrieu and M. Simonovsky. "Large-scale point cloud semantic segmentation with superpoint graphs." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4558–4567.

[77] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun. "Point transformer." In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 16259–16268.