SCHOOL OF COMPUTATION, INFORMATION
AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Robust Techniques to Enhance Label Propagation in Noisy Oversegmented Point Clouds and 2D Images

Fatma Merve Karalı

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Robust Techniques to Enhance Label Propagation in Noisy Oversegmented Point Clouds and 2D Images

Robuste Techniken zur Verbesserung der Label Propagation in Verrauschten, uversegmentierten Punktwolken und 2D Bildern

Author:	Fatma Merve Karalı
Supervisor:	PD Dr. Ing. Habil. Federico Tombari
Advisor:	Lennart Bastian, Stefano Gasperini
Submission Date:	July 17, 2023

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

Munich, July 17, 2023 Fatma Merve Karalı

Acknowledgments

I would like to express my sincere appreciation to my supervisor, PD Dr. Ing. Habil. Federico Tombari, for providing me with the opportunity to conduct my research in this amazing chair. His invaluable guidance has been instrumental in helping me navigate the challenges of this research journey. I am also deeply grateful to my advisors, Lennart Bastian and Stefano Gasperini, for their continuous support, constructive feedback, and critical advice that helped me to shape my research.

I would like to extend my heartfelt thanks to my partner, Mert Kiray, for his unwavering love, support, and encouragement throughout my academic and personal journey. His constant motivation and belief in me have been the driving force behind my success, and I am truly grateful for his presence in my life.

Finally, I would like to acknowledge my family for their love and support throughout my academic journey. Their unwavering encouragement has been the bedrock of my achievements, and I am deeply grateful for their constant support.

Abstract

Semantic segmentation of point clouds plays a crucial role in 3D scene understanding, but obtaining large annotated datasets for this task is time-consuming and error-prone. To address this challenge, weakly supervised learning techniques have been explored, focusing mainly on the 3D domain while neglecting the potential of incorporating complementary information from the 2D domain. In this master's thesis, we adopt an approach that integrates both 2D images and 3D point clouds to enhance weakly supervised point cloud semantic segmentation. By bidirectionally interacting features from these modalities, we leverage the fine-grained texture in 2D images and the geometric information in 3D point clouds to benefit each other. To incorporate additional supervisory signals, we begin by oversegmenting the point clouds and images and assigning unique labels to the resulting supervoxels and superpixels based on our proposed initial label assignment strategy. We then propagate these labels to unlabeled points or pixels within the corresponding supervoxel or superpixel. However, the oversegmented regions suffer from imprecise object boundaries, leading to inaccuracies in the propagated labels and label noise. To address this issue, we introduce a novel noise-robust framework that integrates robust loss functions and innovative loss adjustment strategies. These techniques enhance the learning capacity of the network and enable robust learning with limited annotations. Additionally, we incorporate multi-modality and develop a novel point/pixel-wise confidence calculation algorithm in the oversegmented point clouds and images to obtain reliable labels based on distance metrics. This approach effectively handles challenges related to ambiguous object boundaries and significantly improves the robustness of the framework even with sparse labels. We conduct extensive experiments under various weakly supervised schemes on benchmark datasets, including ScanNetV2 and 2D-3D-S. The results demonstrate that our noise-robust framework outperforms baseline methods both quantitatively and qualitatively, showcasing its effectiveness in addressing the limitations of weakly supervised point cloud semantic segmentation.

Contents

1 Introduction

1.1 Problem Statement and Motivation

The task of 3D scene understanding, which involves extracting semantic and geometric information from a three-dimensional scene, is crucial and challenging in computer vision. Recent advancements in computer vision and deep learning have led to significant progress in 3D scene understanding, enabling applications such as augmented reality, scene modeling, autonomous vehicles, and robotics [1]. Semantic segmentation, a fundamental component of 3D scene understanding, aims to assign semantic labels to each point in a 3D point cloud.

While substantial progress has been made in fully supervised semantic segmentation using large annotated datasets [2, 3, 4], the manual annotation of 3D point cloud datasets remains a time-consuming and error-prone task, presenting a significant challenge. Annotating a single indoor scene dataset, for example, can take approximately 22.3 minutes [5]. Thus, there is an urgent need to reduce the annotation overhead and develop efficient methods for 3D point cloud annotation.

To address the annotation costs, weakly supervised learning techniques have been explored, focusing on utilizing limited supervision for 3D point clouds. Weak supervisory signals, such as point annotations or scene-level and sub-cloud annotations, have been proposed as alternative forms of supervision [6, 7, 8, 9, 10]. However, effective and efficient methods for reducing the annotation overhead in weakly supervised 3D point cloud semantic segmentation remain an open research question.

Label propagation, a crucial technique to enhance the supervision signal in weakly supervised semantic segmentation, has been employed in recent state-of-the-art studies using oversegmentation techniques [11, 12]. However, these approaches primarily leverage 3D data, overlooking the potential of incorporating complementary information from 2D images. In this work, we propose an approach for weakly supervised semantic segmentation of point clouds by oversegmenting both the point clouds and corresponding 2D images. By generating supervoxels and superpixels, we group geometrically related points or pixels and assign initial labels based on a proposed strategy. We then extend these labels to unlabeled points within the supervoxels and superpixels, enriching the training data with additional supervised signals.

It is important to address the inherent noise and lack of well-defined object boundaries in oversegmented point clouds and images. Due to ambiguous object boundaries, points or pixels belonging to different objects may reside within the same supervoxel or superpixel, leading to inaccuracies in propagated labels. The presence of noisy propagated labels poses challenges for deep learning models, which are sensitive to biases introduced during training. Thus, careful consideration and mitigation of the effects of noisy labels are crucial during the training process.

While existing studies have focused on learning with label noise in fully supervised scenarios, mostly in the context of image classification [13, 14, 15], limited work has been done on learning with noisy labels for robust point cloud semantic segmentation [16]. These studies propose techniques such as loss correction, loss reweighting, and label refurbishment.

In our work, we specifically address the problem of label noise in weakly supervised semantic segmentation caused by label propagation in oversegmented point clouds and images with inaccurate object boundaries. To tackle this challenge, we propose a novel robust framework that integrates both 2D images and 3D point clouds, introducing innovative techniques for robust loss adjustment while reducing the annotation overhead.

1.2 Proposed Methodology Overview and Contributions

To the best of our knowledge, this is the first investigation into addressing the issue of noisy labels in oversegmented point clouds and images within the context of weakly supervised semantic segmentation.

To tackle the challenge of noisy labels, we employ a backbone architecture that leverages complementary information from both 2D images and 3D point clouds. By integrating these modalities, our approach captures geometric information from 3D features and color/texture characteristics from 2D features. Additionally, we investigate domain-specific loss functions within our robust framework and introduce novel loss adjustment methods to handle the presence of noisy labels in oversegmented point clouds and images, enhancing the learning capacity of the network and enabling robust learning.

The contributions of this thesis can be summarized as follows:

- We propose a novel robust framework for weakly supervised 3D semantic segmentation that enables the network to learn robust representations from noisy oversegmented point clouds and images.

- We incorporate multi-modality into our framework, enhancing label propagation by leveraging geometric features from 3D data and color/texture information from 2D features. This integration increases the network's resilience to noisy labels.

- We extensively investigate different robust loss functions to handle label noise in oversegmented point clouds and images and improve the network's learning capacity, avoiding biased learning. The most suitable loss function is integrated into our proposed robust framework.
- We introduce an additional level of granularity into the semantic segmentation process through novel loss adjustment methods, specifically loss reweighting. This technique effectively mitigates the negative impact of noisy labels by adjusting the loss of all training examples before updating the neural network. By assigning appropriate weights based on distance metrics, we address challenges such as noise, occlusion, and accurate delineation of complex object boundaries.
- We conduct extensive experiments on the ScanNetV2 [5] and 2D-3D-S [17] datasets, which is a superset of S3DIS [18] under different weakly-supervised annotation strategies. Our benchmark evaluations demonstrate significant improvements over baselines, both quantitatively and qualitatively.

Through our quantitative and qualitative evaluations, we demonstrate that our proposed framework achieves robustness against label noise caused by label propagation in oversegmented point clouds and images with inaccurate object boundaries.

1.3 Thesis Outline

The remaining sections of the thesis are structured as follows:

Chapter 2 introduces the key concepts necessary for understanding the subsequent chapters.

Chapter 3 presents an extensive review of the existing literature in the field. We discuss the state-of-the-art techniques and methodologies related to semantic segmentation, label propagation, robust learning, and handling noisy labels in various domains.

Chapter 4 describes our proposed framework in detail. We present the architecture, including the integration of multi-modal information from 2D images and 3D point clouds, label propagation techniques, and the novel loss adjustment methods employed.

Chapter 5 provides a comprehensive evaluation of our framework, comparing its performance to state-of-the-art methods on benchmark datasets and conducting detailed ablation studies to analyze the contributions of individual components.

Chapter 6 summarizes our findings and contributions, emphasizing the key contributions of our research, and suggests future research directions to advance the field and tackle remaining challenges.

2 Background

This chapter provides an overview of the key concepts necessary for comprehending the subsequent chapters. It introduces and explains fundamental concepts in the fields of semantic segmentation, oversegmentation, robustness, and deep learning with label noise.

2.1 Semantic Segmentation

Semantic segmentation is a fundamental task in the field of computer vision, aimed at accurately dividing images and point clouds into semantically meaningful regions. It holds great significance in various domains, including augmented reality, autonomous driving, and medical image analysis. The primary objective of semantic segmentation is to assign semantic labels, such as "table" or "chair," to individual pixels or points. Extensive research efforts have been dedicated to developing techniques for semantic segmentation in both 2D images and 3D point clouds.

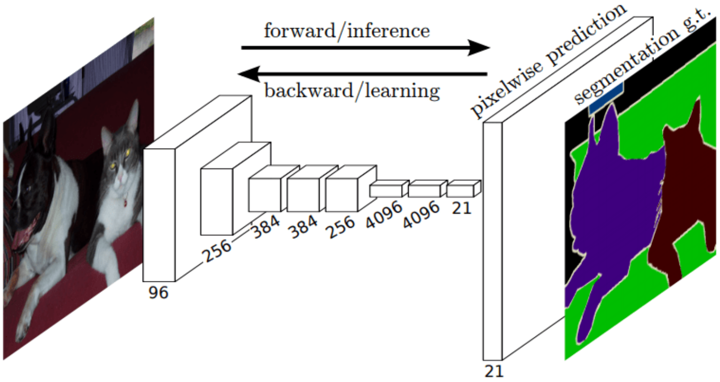


Figure 2.1: A fully convolutional image segmentation network, figure from [19].

In the domain of semantic image segmentation, a significant breakthrough was achieved with the introduction of Fully Convolutional Networks (FCNs) by [19]. FCNs demonstrated the potential of deep learning methods by enabling end-to-end training on images. Unlike traditional networks that employ fully connected layers, FCNs are specifically designed to make dense predictions for per-pixel tasks like semantic segmentation. This is achieved by replacing the final dense layers with convolutional layers, allowing the network to directly output a segmentation map, as depicted in Figure 2.1.

U-Net [20] is another prominent architecture for semantic image segmentation. It adopts an encoder-decoder framework, consisting of an encoder that captures strongly correlated semantic information and a decoder that utilizes additional intermediate layers to propagate this information. By integrating skip connections, the U-Net architecture efficiently recovers spatial details, leading to accurate segmentation results.

In the domain of 3D point cloud semantic segmentation, PointNet [2] was introduced, a groundbreaking method for processing unordered point clouds. PointNet is specifically designed to handle the challenges posed by unordered point cloud data and learn per-point features using shared Multi-Layer Perceptrons (MLPs). Additionally, PointNet incorporates symmetrical pooling functions to capture global features, enhancing the network's ability to capture comprehensive information from the point cloud. Building upon PointNet, several point-based networks have been proposed, utilizing modules like T-Net for point cloud alignment and shared MLPs for per-point feature extraction, as illustrated in Figure 2.2.

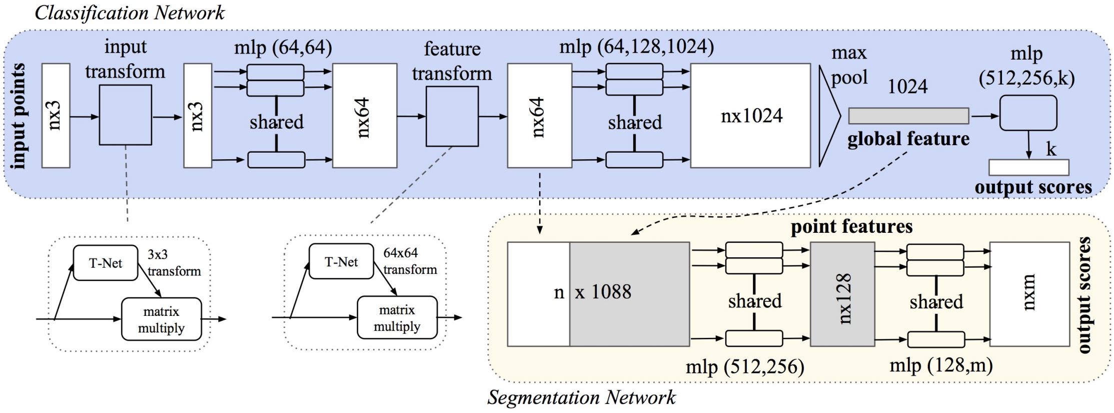


Figure 2.2: PointNet architecture for 3D point cloud semantic segmentation, figure from [2].

It is important to note that while both image semantic segmentation and 3D point cloud semantic segmentation share the common goal of predicting class labels for individual pixels or points, they possess inherent differences in data structure and representation. These differences give rise to unique challenges, necessitating tailored approaches for each domain.

By gaining a comprehensive understanding of the advancements and techniques in semantic segmentation for both images and point clouds, we can leverage this knowledge to develop effective and efficient methodologies for weakly supervised semantic segmentation. This is particularly relevant in scenarios involving noisy oversegmented point clouds and images, where precise and reliable segmentation is of utmost importance.

2.2 Minkowski Convolutional Neural Network

In various real-world applications, such as robotics, virtual reality, augmented images, and medical imaging, the acquisition of 3D scans is crucial. These scans are typically obtained using Light Detection and Ranging (LiDAR) scanners and Magnetic Resonance Imaging (MRI) scanners. However, when dealing with 3-dimensional scans or higher-dimensional spaces, the use of dense representations becomes inefficient due to the sparsity of the data. While certain data, like images, naturally exhibit denseness, other sources, such as 3D point clouds captured by LiDAR scanners or RGB-D cameras, inherently possess sparsity. Point clouds differ from regular images as they are sparse data structures, with most voxels being empty in 3D space. Consequently, traditional "dense" convolutional networks are highly inefficient when applied to such sparse data.

To tackle this challenge, [21] proposed the Minkowski Engine. The Minkowski Engine serves as an auto-differentiation framework that supports sparse tensors, enabling training and evaluation with varying numbers of points in each object.

This work adopts sparse tensors due to their expressiveness and generalizability in high-dimensional spaces. A sparse tensor transforms an input into unique coordinates, associated features, and optionally labels for semantic segmentation during training. This representation extends the concept of a sparse matrix to N-dimensional spaces. In the Minkowski Engine, the Coordinate list (COO) format is employed to store sparse tensors due to its efficiency in neighborhood queries.

In addition to the COO format, the Minkowski Engine introduces various operations, including generalized sparse convolution, sparse tensor quantization, max pooling, global average pooling, sum pooling, and non-spatial functions. The generalized sparse convolution encompasses not only sparse convolutions but also conventional dense convolutions.

The flexibility of the generalized sparse convolution allows for arbitrary strides and kernel shapes, facilitating the creation of high-dimensional networks solely using generalized sparse convolutions. This simplifies implementation and promotes generalizability. Additionally, recent architectural innovations in 2D networks can be directly adopted for high-dimensional networks. For U-shaped variants, the authors add multiple strided sparse convolutions and strided sparse transpose convolutions with skip connections to the base residual network [22], as illustrated in Figure 2.3.

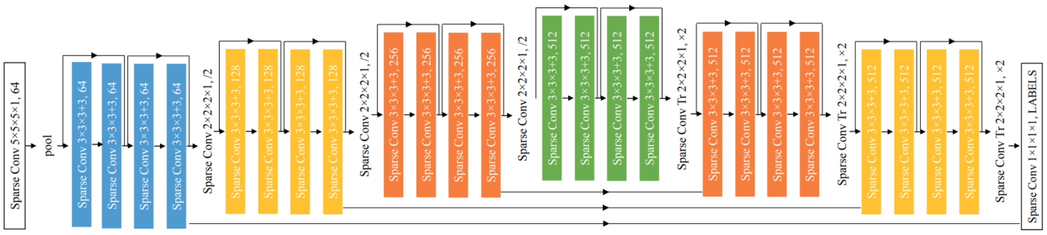


Figure 2.3: Architecture of MinkowskiUNet32, figure from [21].

2.3 Oversegmentation

Oversegmentation is a fundamental technique extensively employed in computer vision, particularly in the domains of image and point cloud semantic segmentation. Its primary objective is to group pixels or points into coherent regions that align with object boundaries. Within the context of oversegmentation, two key concepts are utilized: superpixels for images and supervoxels for point clouds. These concepts serve as compact representations of regions in 2D images and 3D point clouds, respectively.

The utilization of superpixels and supervoxels offers several advantages in the field of computer vision. Firstly, they enable region-based operations, such as feature computation, to be performed on cohesive regions rather than on scattered pixels or points. This region-based approach enhances computational efficiency by reducing the number of primitives that need to be processed, leading to improved performance of vision algorithms. By reducing the complexity of the data representation, superpixels and supervoxels facilitate tasks such as object detection and semantic segmentation [23].

2.3.1 Superpixels

Superpixels are crucial components in image processing, providing a more efficient and meaningful representation of image content through the division of an image into small, homogeneous, and regular regions based on their low-level properties [24]. Superpixels possess desirable properties such as homogeneity, connected partition, and regularity, enabling them to overcome the limitations of pixel discretization and reduce computational complexity.

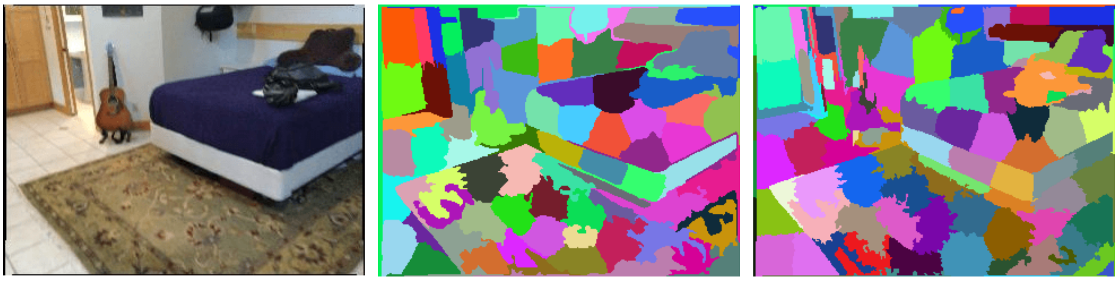
One approach for generating superpixels is through graph-based algorithms. These algorithms treat the image as a planar graph, with pixels serving as vertices and edges representing the connectivity between adjacent pixels. By formulating superpixel generation as a graph-partitioning problem and analyzing the strength of connectivity between pixels, these algorithms effectively partition the graph to obtain cohesive and visually meaningful superpixels.

Another approach for superpixel generation is clustering-based algorithms, which leverage the relative positions of pixels to create cohesive regions. These algorithms employ clustering techniques like k-means to iteratively refine an initial pixel clustering until specific criteria, determined by the algorithm, are met. This iterative refinement process ensures the generation of cohesive superpixels.

One notable clustering-based algorithm for superpixel generation is the Simple Linear Iterative Clustering (SLIC) algorithm [25]. SLIC is an efficient method that adapts the k-means algorithm to oversegment an image into a regular grid. By computing average color and position features for each superpixel and iteratively reassigning pixels to the most similar superpixel, SLIC achieves connected and visually coherent superpixels. A post-processing step further ensures the connectedness of disjoint pixel sets. SLIC offers linear complexity based on the number of pixels and employs a distance metric that combines spatial position and intensity information, resulting in compact superpixels that adhere well to image contours. Its computational speed and ability to produce high-quality results make SLIC suitable for large-scale image analysis and high-resolution models.

In recent years, deep learning methods have shown promise in developing supervised superpixel oversegmentation approaches. One such method is the Superpixel Sampling Network (SSN) [26], which provides an end-to-end trainable solution for learning task-specific superpixels. SSN addresses the challenge of non-differentiability in existing superpixel algorithms by proposing a differentiable algorithm based on SLIC. By relaxing the nearest neighbor constraints present in SLIC, the modified algorithm enables end-to-end training and leverages the power of deep networks for learning superpixels. SSN combines the pixel-wise features obtained from the deep network with the differentiable SLIC, allowing iterative clustering to generate the desired superpixels. This approach facilitates the utilization of flexible loss functions and ensures efficient runtime, offering a novel solution for supervised superpixel oversegmentation.

The field of superpixel generation encompasses various approaches, including graph-based and clustering-based algorithms. SLIC and SSN are two prominent methods that have demonstrated effectiveness and efficiency in producing high-quality superpixels while ensuring adherence to object boundaries. Figure 2.4 presents the visual results of SLIC and SSN applied to a 2D image from ScanNetV2 dataset [5].



2D Image SLIC [25] SSN [26]

Figure 2.4: Comparison results for different superpixel algorithms applied to a 2D image from the ScanNetV2 dataset [5], showcasing their performance in generating superpixel representations.

2.3.2 Supervoxels

Supervoxels are small regions composed of perceptually similar voxels within 3D point clouds, similar to the concept of superpixels in 2D images. The task of oversegmenting 3D point clouds into supervoxels poses a significant challenge due to the unordered nature and irregular distribution of points in 3D space [27].

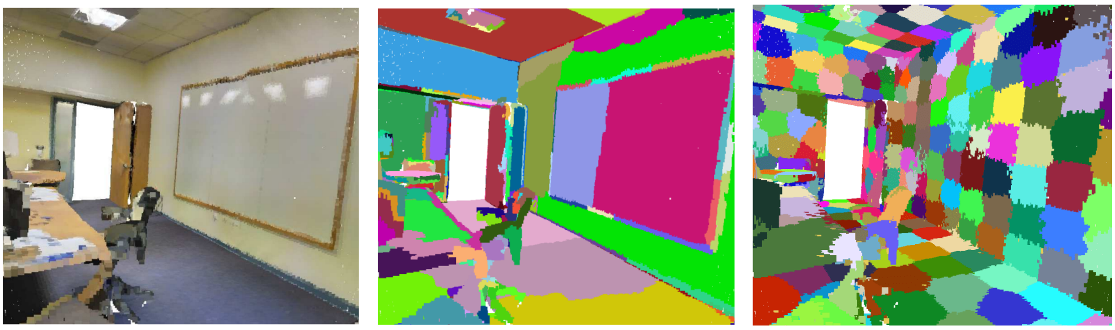
To ensure effective generation of supervoxels in 3D point clouds, several properties are considered. Spatiotemporal uniformity, also referred to as conservatism, aims to create compact and uniformly shaped supervoxels in both spatial and temporal dimensions, maintaining consistency and regularity within the supervoxel regions [27].

Another crucial property is spatiotemporal boundaries and preservation. It is important for supervoxel boundaries to align with object or region boundaries when present and remain stable in the absence of clear object boundaries. This ensures that supervoxels accurately capture the spatial and temporal boundaries of objects or regions within the point cloud [27].

Moreover, the oversegmentation process into supervoxels should not compromise the overall performance of the application. Maintaining the desired level of accuracy and performance is essential during the division of the point cloud into supervoxels, ensuring that the supervoxel representation does not degrade the system's performance [27].

In the context of oversegmenting point clouds into supervoxels, several studies have been conducted. The Voxel Cloud Connectivity Segmentation (VCCS) [28] and Boundary-Enhanced Supervoxel Segmentation (BESS) [29] algorithms are prominent works that leverage the 3D geometry of the scene. VCCS employs a cluster-based method based on the k-means algorithm and octrees for point cloud voxelization as initial steps. On the other hand, BESS proposes a two-stage supervoxel oversegmentation approach, incorporating a graph-structured method to preserve object shapes and a boundary detection technique.

While VCCS and BESS rely on hand-crafted geometric features, a deep learning-based approach called Supervized SuperPoint (SSP) [30] presents a supervised framework for oversegmenting 3D point clouds. SSP utilizes a lightweight neural network to learn deep embeddings of local geometry, emphasizing high contrast at object boundaries. These embeddings are computed based on the local neighborhood of points, and the point cloud oversegmentation is formulated as a graph partition problem using the learned embeddings. In our methodology, we employed VCCS as an unsupervised and efficient method for oversegmenting point clouds in a weakly supervised setting. The oversegmentation results for VCCS and SSP can be observed in Figure 2.5.



Original point cloud SSP [30] VCCS [28]

Figure 2.5: Comparison results for different supervoxel algorithms, including SSP and VCCS, applied to a 3D point cloud from the ScanNetV2 dataset [5], illustrating their effectiveness in generating supervoxel representations.

2.4 Robustness

Robustness, as defined by the [31], refers to the degree to which a system or component can operate correctly in the presence of invalid inputs or challenging environmental conditions. In the computer vision field, robustness encompasses various aspects, including the ability to maintain performance on manipulated or modified inputs, generalize across different domains, and resist adversarial attacks [32].

Robustness is a relative measure of model performance rather than an absolute one. When considering robustness, it is essential to take into account the characteristics of data corruption, the design and optimization of the model to mitigate such corruptions, and the evaluation methods used to assess performance [32].

In the context of deep learning methods for computer vision, robustness plays a crucial role in evaluating model performance and ensuring their ability to handle various factors that can impact predictions. One such factor is adversarial attacks, where neural networks trained on specific datasets can be deceived by inputs that are subtly different from the training data [33].

Another factor that affects the robustness of deep learning models is label noise, which arises from misclassification of labels. Label noise can arise from various sources, such as distributional shifts, errors in data entry, insufficient data descriptions for class labeling, decisions made by non-experts, and instances that lie near the boundaries of different classes [34].

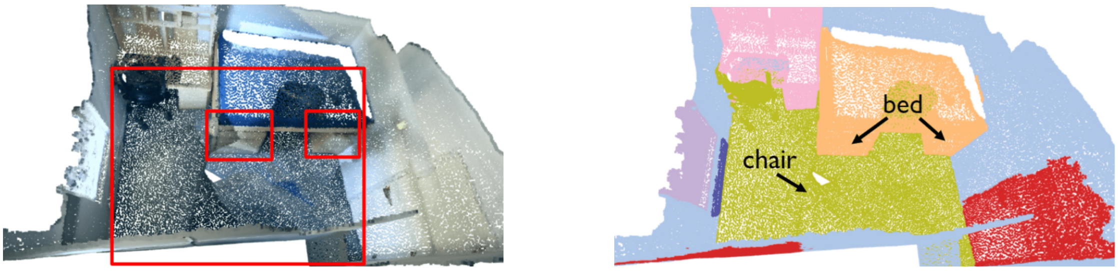
By addressing these factors, deep learning models can be evaluated and designed to be robust to these challenging conditions, ultimately enhancing their practical utility and reliability in real-world applications.

2.5 Deep Learning with Label Noise

Deep learning with label noise is an important aspect to consider in the context of semantic segmentation. In the task of semantic segmentation, ground truth labels play a crucial role as they provide the model with the necessary information to learn the probability of assigning each point/pixel in a point cloud/image to a specific class. While fully supervised scenarios with rich ground truth labels are ideal, the process of obtaining accurate ground truth labels is prone to human errors, leading to potential label noise and subsequent impact on the model's performance. Moreover, accurate labeling requires domain expertise, as observed in fields like medical imaging and scene understanding, and can be time-consuming. Additionally, since different experts may have varying interpretations, it is often necessary to reach an agreement on the labels or combine annotations from multiple sources, which can be costly [35].

Labeling 3D datasets poses additional challenges compared to 2D images. The larger number of points in 3D datasets requires extensive labeling efforts, and the dynamic nature of 3D geometry, including changing views, positions, and scales, further complicates the annotation process [16]. Annotators must possess specialized expertise and a comprehensive understanding of the structure to accurately label 3D data. As a result, 3D labeling is more susceptible to errors, leading to label noise as demonstrated in Figure 2.6, where ScanNetV2 [5] exhibits label noise, such as mislabeling the floor as a chair.

In the context of sparse label settings, where only a limited number of ground truth labels are available, the impact of label noise becomes even more noticeable compared to fully supervised scenarios. To address the challenge of limited labels, state-of-the-art weakly supervised segmentation models [11, 12] often utilize oversegmentation tech-



Input Scenes

Noisy Ground Truth Labels

Figure 2.6: Illustration of the label noise in the context of point cloud semantic segmentation on the ScanNetV2 dataset [5]. The input scenes show noisy instances (highlighted with red boxes), and the ground truth semantic annotation exhibits label noise, such as mislabeling the floor as a chair or mislabeling a cabinet as a bed, figure from [16].

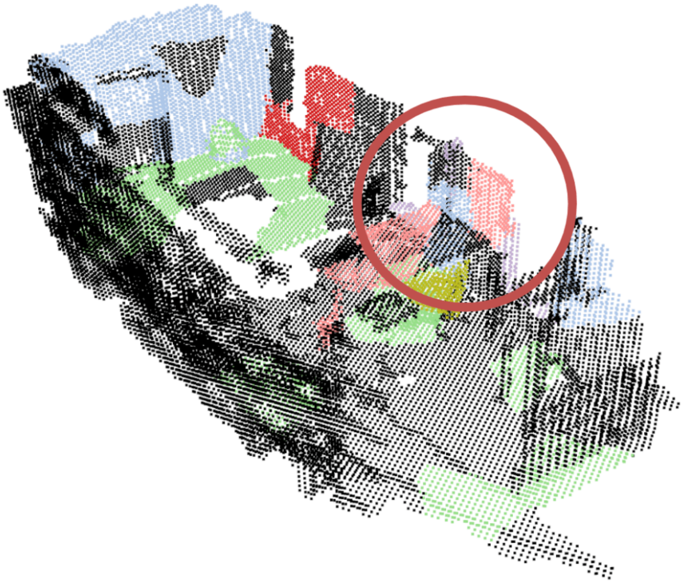


Figure 2.7: Illustration of label noise arising from oversegmentation in the context of sparse label settings on the ScanNetV2 dataset [5]. The red circle highlights an example where the unknown region is mistakenly labeled as a table instead of a wall, emphasizing the presence of label noise in oversegmented point cloud.

niques. These techniques aim to propagate the sparse labels to supervoxels, providing larger labeled regions for training. However, it is important to note that oversegmentation may not consistently align with object boundaries, leading to label errors within the oversegmented regions, as illustrated in Figure 2.7. Such label noise within supervoxel boundaries can significantly impact the accuracy of segmentation results, particularly when objects are wrongly labeled within the supervoxel.

Nevertheless, striking a balance between effectively handling label noise and preserving the discriminative information within labeled data is crucial. Recognizing the detrimental impact of label noise, robust techniques have been proposed to enhance the model's resilience to label noise and mitigate its negative effects. These techniques aim to ensure that the propagated labels accurately capture the underlying semantics of the scene while reducing the influence of label noise. By incorporating robust strategies, deep learning models can overcome the challenges posed by label noise, ultimately improving the accuracy and reliability of semantic segmentation results. These robust strategies can be categorized into five different methods [36]: robust architecture, robust regularization, robust loss function, loss adjustment, and sample selection.

2.5.1 Robust Architecture

The concept of robust architecture is focused on developing training methods that enhance the resilience of deep neural networks against label noise. The primary objective is to introduce architectural modifications to the neural network, such as the incorporation of a noise adaptation layer, to achieve this goal effectively.

One prominent approach in implementing a robust architecture involves the utilization of a noise layer, as proposed by [37]. This layer generates a transition matrix that captures the relationship between noisy and true labels, facilitating the modeling of the noise transition process. The noise adaptation layer expresses the posterior probability of a noisy class as a weighted sum of the posterior probabilities of the true classes, where the weights are determined based on the noise transition matrix.

Researchers have explored various methods for integrating it with different network architectures. For instance, [38] proposed the Robust Conditional GAN (RCGAN), which incorporates a noise layer within a generative adversarial network (GAN) framework. This integration harnesses the power of GANs in generating realistic data while leveraging the noise adaptation layer's ability to handle label noise, resulting in improved robustness.

Another notable study by [39] introduced an expectation-maximization (EM) algorithm specifically designed to optimize the parameters of the noise layer. By refining the performance of the noise adaptation layer, this algorithm enhances its capacity to accurately model the noise transition process.

2.5.2 Robust Regularization

Robust regularization encompasses a range of techniques designed to enhance the generalizability of deep neural networks when faced with label noise. These techniques can be classified into two categories: explicit and implicit regularization methods.

Explicit regularization techniques are specifically designed to constrain the effective capacity of a model, thereby improving its generalization performance [40]. Weight decay [41], is an example of explicit regularization. It reduces overfitting in feed-forward neural networks by penalizing large weight values. By choosing the smallest weight vector that solves the learning problem, weight decay effectively suppresses irrelevant components of the weight vector, promoting improved generalization.

Another explicit regularization technique is dropout [42]. During training, dropout randomly deactivates units, reducing the network's sensitivity to individual noisy labels. By explicitly controlling the model's capacity, dropout helps prevent overfitting to noisy labels, leading to enhanced robustness.

Implicit regularization, on the other hand, achieves regularization effects without explicitly modifying the loss function or constraining the model's capacity [40]. Data augmentation [43], is an example of an implicit regularization technique. It introduces random transformations, such as rotations or translations, to the training data, increasing its diversity. This augmented data helps the model learn more robust and generalized representations, improving its ability to handle label noise and unseen data.

Mini-batch stochastic gradient [44] is another implicit regularization technique widely used in training deep neural networks. It involves randomly sampling a subset of the training data for each iteration, introducing stochasticity into the optimization process. This randomness prevents the model from overfitting to specific examples and promotes generalization.

By combining both explicit and implicit regularization techniques, researchers aim to enhance the robustness and generalizability of deep neural networks in the presence of label noise. These techniques provide effective mechanisms to control the model's capacity, increase data diversity, and introduce stochasticity into the training process, ultimately improving the network's ability to handle label noise and generalize well to unseen data.

However, as depicted in Figure 2.8, the application of these regularization techniques alone does not sufficiently improve test accuracy. A notable accuracy difference persists between models trained with noisy labels and those trained with clean labels.

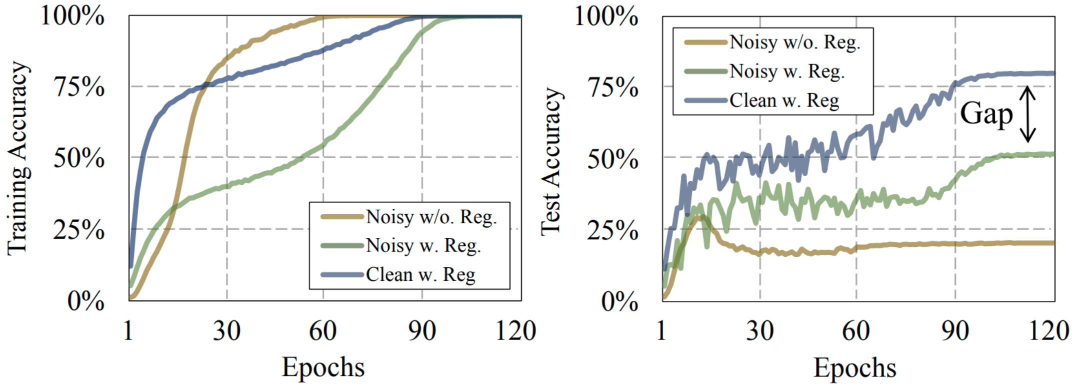


Figure 2.8: Comparison of convergence curves for training and test accuracy of a classification network on the CIFAR-100 dataset [45], showcasing the impact of regularization techniques on models trained with noisy and clean labels. The curves demonstrate the performance of models trained without regularization on noisy data ("Noisy w/o Reg."), models trained with regularization on noisy data ("Noisy w. Reg."), and a model trained with regularization on clean data ("Clean w. Reg."), figure from [36].

2.5.3 Robust Loss Functions

In addition to robust architecture and robust regularization techniques, the choice of a robust loss function provides another approach to effectively handle label noise during the training of deep neural networks.

The loss function plays a fundamental role in guiding the learning process of a deep neural network by minimizing the discrepancy between its predictions and the ground truth labels. In the presence of label noise in the training dataset, employing a robust loss function becomes desirable over non-robust alternatives.

A robust loss function is specifically designed to be less sensitive to outliers or noisy data points present in the training set. It aims to mitigate the adverse effects that extreme values may have on the learning process and the overall performance of the model.

In the context of semantic segmentation, the selection of an optimal loss function for a network does not follow a fixed rule, as it heavily depends on the network architecture and the characteristics of the input data.

The Dice loss [46], Focal loss [47], and Tversky loss [48] are examples of robust loss functions commonly employed in semantic segmentation tasks. These loss functions exhibit robustness by effectively handling class imbalance, capturing spatial context, adapting loss contributions based on prediction confidence, and allowing control over the trade-off between false positives and false negatives.

2.5.4 Loss Adjustment

In addition to robust architecture, robust regularization techniques, and robust loss functions, loss adjustment methods provide another effective strategy to mitigate the influence of label noise during the training of deep neural networks. These methods aim to reduce the negative impact of noisy labels by modifying the loss of training examples before updating the neural network parameters [36]. Loss adjustment methods can be categorized into four groups: loss correction, loss reweighting, label refurbishment, and meta-learning.

1. **Loss Correction:** Loss correction methods estimate the noise transition matrix and incorporate it into the loss computation for each training example, thereby adjusting the loss [36]. By considering the noise characteristics, these methods effectively account for label noise and mitigate its influence during the learning process. The noise transition matrix provides insights into the probabilities of label transitions and guides the adjustment of the loss values, leading to improved robustness to label noise.

2. **Loss Reweighting:** Loss reweighting methods assign different levels of importance or confidence to each training sample based on the likelihood of having correct labels [36]. The objective is to down-weight samples that are more likely to have incorrect labels while assigning greater weights to examples with true labels. This approach acknowledges the presence of label noise and aims to give more emphasis to reliable examples, thereby reducing the impact of noisy instances. By adjusting the loss weights accordingly, these methods promote learning from trustworthy samples and improve the robustness of the training process. In our methodology, we also introduce various algorithms for loss reweighting to enhance the resilience to label noise.

3. **Label Refurbishment:** Label refurbishment methods modify the loss by incorporating the refurbished label, which is obtained through a convex combination of the noisy label and the predicted label [36]. Instead of directly using the noisy label for loss computation, this approach backpropagates the loss computed using the refurbished label, which incorporates the model's prediction.

4. **Meta-learning:** Meta-learning is a loss adjustment approach that focuses on automatically inferring the optimal rule for loss adjustment [36]. It involves

learning to learn at a higher level, which goes beyond conventional learning. Meta-learning aims to discover data-agnostic and noise type-agnostic rules for loss adjustment.

2.5.5 Sample Selection

Sample selection strategies serve as a robust solution to address the challenges arising from label noise in training datasets. These strategies aim to identify true-labeled examples while mitigating the inclusion of corrupted labels by leveraging collaborative efforts from multiple networks or adopting multi-round learning techniques [36].

However, it is crucial to acknowledge that although learning with sample selection is a well-motivated approach with proven effectiveness, it is not enduring to the accumulation of errors stemming from incorrect selection, particularly in the presence of datasets comprising numerous ambiguous classes.

3 Related Work

This chapter provides a comprehensive review of the current literature in the field. We explore the latest techniques and methodologies concerning semantic segmentation, label propagation, robust learning, and handling noisy labels across different domains.

3.1 Fully Supervised Semantic Segmentation

Semantic segmentation is a fundamental and challenging task in scene understanding, aiming to assign class labels to each pixel or point in an image or point cloud. Fully supervised semantic segmentation, as a widely explored concept in this domain, has attracted considerable attention in academic research and has been applied to various practical applications.

Fully supervised semantic segmentation encompasses a diverse range of methodologies that operate on different types of input data. One common approach focuses on RGB data, where deep learning architectures are trained to classify individual pixels in 2D images with class labels. These methods leverage convolutional neural networks (CNNs) [49] to extract informative features from RGB images, enabling pixel-level classification.

Another avenue of exploration within fully supervised semantic segmentation involves the use of point cloud data. Point clouds, obtained from 3D sensors, provide a rich representation of the environment and enable the inference of fine-grained object boundaries. Deep learning techniques tailored for point cloud semantic segmentation have been developed to accurately segment objects and scenes in three-dimensional space, thereby extending the scope of semantic segmentation to 3D data.

Furthermore, there is a growing interest in combining RGB and depth information, commonly referred to as RGB-D data, for semantic segmentation tasks. These approaches leverage the complementary nature of RGB and depth data, enabling the extraction of both appearance-based and geometric cues for improved segmentation accuracy. By jointly analyzing RGB and depth information, these methods aim to capture richer contextual information and enhance the understanding of complex scenes.

In addition to single-modal data analysis, fully supervised semantic segmentation also extends to multi-modal data fusion. This involves the integration of data from multiple sources, such as RGB images, point clouds, and depth maps, to achieve a comprehensive understanding of the scene. By fusing information from different modalities, these techniques aim to exploit the strengths of each modality and mitigate their individual limitations, ultimately improving the quality of semantic segmentation results.

The exploration of fully supervised semantic segmentation methods across various data modalities highlights the ongoing research efforts in the field of scene understanding. By leveraging RGB, point cloud, RGB-D data, and incorporating multi-modal fusion techniques, researchers strive to advance the state-of-the-art in semantic segmentation and enable more accurate and robust scene interpretation.

3.1.1 Fully Supervised 2D Semantic Segmentation

2D semantic segmentation involves the task of assigning class labels to individual pixels in 2D images. Several notable approaches have been proposed in the literature, leveraging deep learning techniques to achieve accurate and efficient pixel-wise segmentation.

One influential method in this field is FCNs [19] , which introduced the concept of converting classification networks, such as AlexNet[50] and VGGNet [51], into fully convolutional networks. By replacing fully connected layers with convolutional layers, FCNs enable end-to-end training for pixel-to-pixel semantic segmentation, preserving spatial information and producing dense segmentation maps.

The Deconvolution Network [52] proposed a semantic segmentation algorithm that utilizes a deconvolution network trained on top of a VGG-based convolutional network. This approach focuses on recovering spatial information and refining object boundaries through deconvolutional layers. Similarly, U-Net [20] and SegNet [53] introduced encoder-decoder architectures for 2D semantic segmentation, enabling the learning of both low-level and high-level features to extract fine-grained details.

To address the issue of poor localization properties in deep networks, DeepLab [54] combined deep convolutional neural networks with a fully connected conditional random field (CRF). This integration improved the localization accuracy of segmentation results by capturing long-range dependencies. Moreover, dilated convolutions, as demonstrated by [55], have been employed to aggregate multi-scale contextual information without sacrificing resolution or coverage. By allowing the network to access a broader context, these dilated convolutions enhance segmentation accuracy while preserving spatial details.

Global context information has also been leveraged to improve the performance of 2D semantic segmentation. Methods like ParseNet [56] and PSPNet [57] exploit the capability of global context information to enhance segmentation accuracy. Attention mechanisms have emerged as a powerful tool for capturing long-range contextual information in a flexible and adaptive manner. For example, EncNet [58] and CCNet [59] utilize attention mechanisms to dynamically weight the contributions of different spatial locations based on their contextual relevance, effectively enhancing the discriminative power of the network.

3.1.2 Fully Supervised 3D Semantic Segmentation

The field of 3D semantic segmentation has witnessed significant advancements, with various methods categorized based on the data representations they employ. These categories include point-based methods, voxel-based methods, and 2D-projection-based methods, each offering unique approaches to tackle the challenges of 3D scene understanding.

Point-based methods, PointNet [2], PointNet++ [60], leverage MLPs architectures for 3D scene understanding by directly operating on the individual points of a point cloud. To enhance the capabilities of point-based methods, convolution-based approaches, including PointCNN[4], KPConv [61], PointConv [62], and FPConv [63], introduce convolution operations tailored for point cloud data, enabling effective local feature extraction.

Another category of point-based methods focuses on enhancing local region features. Examples of such methods include SpiderCNN [64], DGCNN [65], PointWeb [66], and RandLA-Net [67]. These approaches leverage local neighborhoods and hierarchically aggregated features to capture fine-grained details in point clouds.

Attention-based aggregation techniques have also been employed in point-based methods to improve semantic segmentation performance. Methods such as Attentional ShapeContextNet (A-SCN) [68], PCAN [69], and Point Attention Transformers (PATs) [70] utilize attention mechanisms to dynamically weight the contributions of different points based on their contextual relevance, enhancing the discriminative power of the network.

Graph construction methods have also been explored in fully supervised 3D semantic segmentation. Graph Attention Convolution (GAC) [71], and Hierarchical Point-Edge Interaction Network [72] are examples of techniques that leverage graph structures to model relationships between points. However, despite their promising results, these point-based methods face challenges in directly scaling to large scenarios due to their high computational and memory requirements.

Voxel-based networks have emerged as an efficient approach for handling large-scale point cloud data by discretizing the 3D space into regular voxel grids. Notable voxel-based methods include VoxNet [73], SegCloud [3], and OctNet [74]. Voxel grids offer a structured representation and enable the application of 3D convolutions. However, voxel-based methods may suffer from empty voxels due to the sparsity of point clouds, resulting in redundant computations and limited efficiency.

Another approach in 3D semantic segmentation is 2D-projection-based methods, which leverage the multi-view mechanism by projecting unstructured 3D points onto 2D images captured from different camera views. These methods exploit the rich information present in 2D images and perform semantic segmentation in the projected space. Various methods have been developed in this category, including Multi-view CNN [75], Volumetric and Multi-View CNNs [76], and SalsaNet [77]. Compared to voxel and point-based approaches, 2D projection methods offer more compact and dense representations, enabling real-time computations, but they may suffer from information loss caused by the projection process.

Recent advancements in fully supervised 3D semantic segmentation have been made by Submanifold Sparse Convolutional Networks (SSCN) [78] and OccuSeg [79]. These methods employ sparse 3D voxel grids and utilize sparse 3D convolutions to extract features. They excel in recognizing 3D patterns and demonstrate strong performance for objects with distinctive 3D shapes, such as chairs. However, they may face challenges when dealing with other types of objects, such as pictures, and require significant memory resources, which limit spatial resolutions and batch sizes. Notably, Minkowski Convolutional Neural Networks [21] introduce a novel 4D sparse convolution approach for spatio-temporal 3D point cloud data, providing an open-source library that supports auto-differentiation for sparse tensors. This approach stands out in terms of both accuracy and efficiency, achieving state-of-the-art results.

Despite the promising outcomes, scaling these methods to large scenarios remains challenging due to their high computational and memory requirements. Additionally, the lack of detailed texture and color information in these methods may result in limited performance when distinguishing objects with similar appearances.

3.1.3 Fully Supervised Semantic Segmentation Recognition with combined 2D-3D data

Improving the performance of semantic segmentation in both 2D and 3D domains has been the focus of several studies that aim to fuse information from both modalities. By leveraging the strengths of 2D appearance information and 3D geometric relations, these approaches enhance scene understanding and provide more detailed geometric information about objects. In this section, we discuss notable works that utilize combined 2D-3D data for semantic segmentation.

One approach proposed [80] introduces a 3D graph neural network for RGB-D semantic segmentation. This method builds a k-nearest neighbor graph on top of the 3D point cloud, allowing joint reasoning about the data by considering both 2D appearance information and 3D geometric relations. By combining these modalities, the model achieves improved performance in recognizing object categories and segmenting them in 3D space.

To incorporate depth information into the CNN architecture and improve semantic segmentation, Depth-aware CNN (D-CNN) [81] presents a depth-aware network for RGB-D Segmentation. This approach leverages both RGB and depth data, leading to enhanced recognition accuracy and providing more detailed geometric information about objects in the scene.

An end-to-end convolutional neural network that combines RGB input and 3D geometry, 3DMV [82] is proposed. This network backprojects multi-view 2D features to 3D volumes and predicts dense semantic labels on a voxel grid. The joint utilization of 2D and 3D data significantly improves the accuracy of 3D segmentation compared to existing methods.

Another framework, Multi-View PointNet (MVPNet) [83], fuses 2D multi-view images and sparse point clouds in canonical 3D space. This method employs a point-based network to predict 3D semantic labels. By leveraging complementary features and effectively handling occlusions, MVPNet outperforms prior point cloud-based approaches in the task of 3D semantic segmentation.

Additionally, Supervoxel-CNN [84] recognizes that on-surface supervoxels provide a compact representation of 3D surfaces. As a result, they explore a supervoxel-based convolutional neural network, enabling joint 2D-3D learning for 3D semantic prediction. By directly applying a convolution operation on supervoxels, the model effectively incorporates both 2D appearance and 3D geometric information.

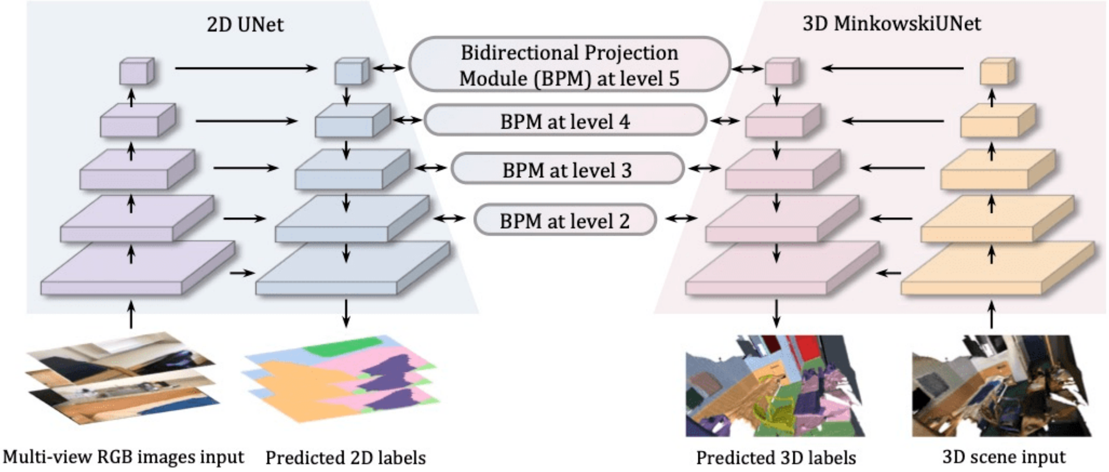


Figure 3.1: Overview of the Bidirectional Projection Network (BPNet) [85].

Bidirectional Projection Network (BPNet) [85] is another significant research contribution that introduces a bidirectional projection network for joint 2D and 3D reasoning in an end-to-end manner. The primary objective of this approach is to leverage the complementary information present in both 2D images and 3D point clouds, enabling their interaction at multiple architectural levels. By combining these two domains, BPNet achieves improved performance in scene recognition.

In our methodology, we adopt the BPNet architecture as the backbone network to exploit the benefits of both the 2D and 3D domains, resulting in a robust framework. By leveraging the detailed texture, color information from 2D images, and the valuable geometric knowledge from 3D point clouds, BPNet enables enhanced scene understanding and superior segmentation performance. The key components of the BPNet architecture, illustrated in Figure 3.1, provide a comprehensive understanding of its functionality and contributions.

The BPNet methodology involves voxelizing 3D point clouds into volumes and feeding them into the 3D sub-network, which is the 3D MinkowskiUNet [21]. Simultaneously, multi-view 2D images are fed into the 2D sub-network, which is the 2D UNet [20]. During training, three random 2D views are sampled to ensure data diversity, while during testing, the 2D frames are divided into three groups, with one central view selected per group to reduce overlap.

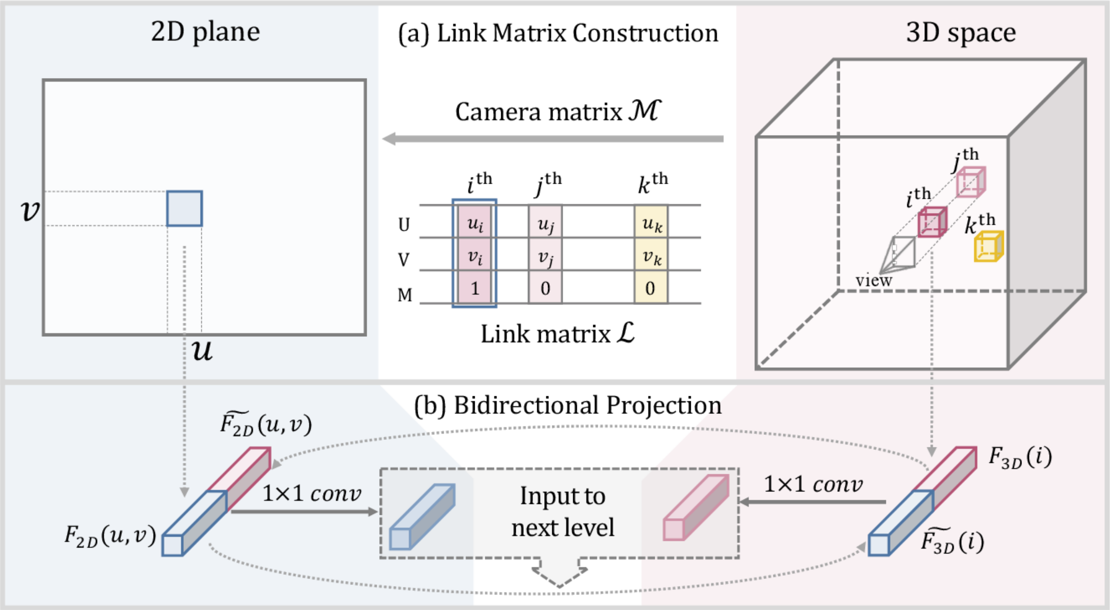


Figure 3.2: Bidirectional Projection Module (BPM) [85].

An essential module within BPNet is the bidirectional projection module (BPM), which establishes bidirectional connections between the 2D and 3D sub-networks within the decoder, enabling the integration of features from both domains and enhancing scene understanding. As depicted in Figure 3.2, the BPM constructs a link matrix that maps voxels to pixels through perspective projection. This link matrix facilitates the projection of 3D features into 2D space and the back-projection of 2D features into 3D space at multiple decoder levels. To combine the projected features with the original features, a 1x1 convolution is applied for fusion. The resulting fused features are then passed to subsequent levels for further processing and refinement.

3.2 Weakly Supervised Semantic Segmentation

The field of weakly supervised semantic segmentation has emerged as a prominent research area in response to the challenges associated with acquiring dense annotations, including their high cost, labor-intensive nature, and potential annotation errors. Fully supervised segmentation networks heavily rely on dense annotations, which are limited and expensive to obtain. This limitation significantly hinders the scalability and practicality of these models. Notably, datasets such as Microsoft COCO [86] require up to 15 times more effort to annotate segmentation masks compared to object locations. Similarly, the annotation process for ScanNetV2 [5] is time-consuming and error-prone, with an average annotation time of approximately 22.3 minutes per indoor scene dataset [11].

To address these challenges and enhance the robustness of semantic segmentation models, recent studies have shifted their focus towards weakly supervised semantic segmentation. This approach aims to overcome the limitations of dense annotations by exploring alternative strategies that require less annotation effort while still achieving reliable segmentation results. By leveraging weak supervision, these methods enhance the generalizability and scalability of semantic segmentation models.

In this section, we conduct a comprehensive review of previous studies conducted in the domains of 2D, 3D, and 2D-3D joint semantic segmentation, all of which specifically address the weakly supervised setting. By examining these studies, we aim to gain valuable insights into the potential of weakly supervised semantic segmentation methods in reducing annotation burdens while ensuring robust segmentation performance. These approaches offer promising directions for effectively utilizing limited annotations and improving the efficiency and adaptability of the segmentation process across various domains.

3.2.1 Weakly Supervised 2D Semantic Segmentation

In the pursuit of effectively utilizing limited annotations, researchers have developed innovative techniques that leverage weakly supervised approaches for 2D semantic segmentation. These methods harness diverse sources of supervision, including image-level labels, bounding boxes, and scribbles, to guide the segmentation process.

Among the various weakly supervised approaches in 2D semantic segmentation, deep CNNs have been widely employed. Several studies have explored the use of image-level labels for segmentation such as MIL-FCN [87], Seed, Expand, and Constrain (SEC) [88], and Superpixel Pooling Network (SPN) [89]. Image-level labels provide an efficient setting, as each training image is assigned a class label indicating the presence of objects belonging to that class within the image. However, accurately associating these image-level labels with their corresponding pixels poses a challenge, as the specific object locations within the image are unknown. To establish pixel-label correspondence, classification activation maps (CAM) [90] was introduced, which identifies the most discriminative regions within the image and uses them as pixel-level supervision for segmentation networks. However, CAM may struggle to capture small and sparse discriminative regions and may not encompass the entire object region.

To address these limitations, researchers have proposed extensions and enhancements to the CAM-based approach. Studies such as [91, 92, 93] have expanded upon the CAM methodology to improve the accuracy of pixel-label assignment. These works aim to address challenges related to small, sparse, or incomplete discriminative regions, contributing to the robustness and reliability of weakly supervised 2D semantic segmentation techniques.

While image-level labels offer efficient supervision, other weakly supervised approaches employ annotations in the form of bounding boxes, such as BoxSup [94], and DeepCut [95], or scribbles, as seen in ScribbleSup [96]. However, these methods often require some degree of human intervention during the annotation process, making them more costly and less scalable for large-scale visual datasets [93].

3.2.2 Weakly Supervised 3D Semantic Segmentation

This section presents several notable approaches that address weakly supervised 3D semantic segmentation, highlighting their contributions and limitations.

[6] propose a framework that leverages incomplete supervision and inexact supervision branches, along with a subsequent smooth branch, to achieve competitive performance with weak supervision. The incomplete supervision branch utilizes annotations that are uniformly distributed in the point cloud, which can pose challenges during the annotation process [11]. However, it should be noted that this framework may encounter memory issues when applied to large-scale point clouds due to the parameter-free graph usage for post-processing [97].

To address the limitations of the previous method, Perturbed self-distillation (PSD) [97] framework for weakly supervised 3D semantic segmentation was introduced. PSD incorporates a self-distillation mechanism to establish consistency between the original point cloud and its perturbed version, enhancing the robustness of the segmentation results. By integrating perturbations into the learning process, PSD effectively leverages weak supervision to improve the overall segmentation accuracy.

Another noteworthy approach is Weakly-supervised framework for Point cloud Recognition (WyPR) [98], which jointly learns semantic segmentation and object detection for point cloud data using only scene-level class tags as supervision. This method addresses the challenge of limited annotations by exploiting scene-level class information and achieves promising results in weakly supervised 3D semantic segmentation. By leveraging the inherent correlation between semantic segmentation and object detection, WyPR demonstrates the potential for jointly learning these tasks with weak supervision.

Graphical information gain-based attention network (GaIA) [99] is another approach proposed for weakly supervised point cloud semantic segmentation. GaIA aims to reduce epistemic uncertainty by employing graphical information gain and the anchor-based additive angular margin loss, ArcPoint [99]. The attention network enables the network to embed unlabeled points with high entropy toward the reliable labeled points, contributing to enhanced segmentation results.

[11] propose the One Thing One Click (OTOC) annotation strategy, where annotators label one point per object. They introduce a self-training approach with iterative training and label propagation, incorporating a graph propagation module and a relation network to model node similarity and generate pseudo labels. Similar to our methodology, they employ oversegmentation of point clouds into supervoxels and expand their OTOC annotations using the supervoxel partition, generating initial pseudo labels that guide subsequent updates. Their framework is detailed in Figure 3.3. Furthermore, [11] adopt a different strategy for oversegmenting point clouds compared to our approach. They use the provided segments from the ScanNetV2 [5] dataset for their ScanNetV2 experiments, while relying on the geometrical partitioning results by [100] for supervoxel partitioning in the S3DIS [17] dataset.

These studies highlight the ongoing efforts to develop effective techniques for weakly supervised 3D semantic segmentation. While each approach presents its unique contributions, challenges such as establishing annotation consistency, memory limitations, and limited supervision sources remain to be addressed.

3.2.3 Weakly Supervised Semantic Segmentation with Combined 2D-3D Data

The exploration of complementary features between the 2D and 3D domains has been extended to weakly supervised semantic segmentation, where robust segmentation

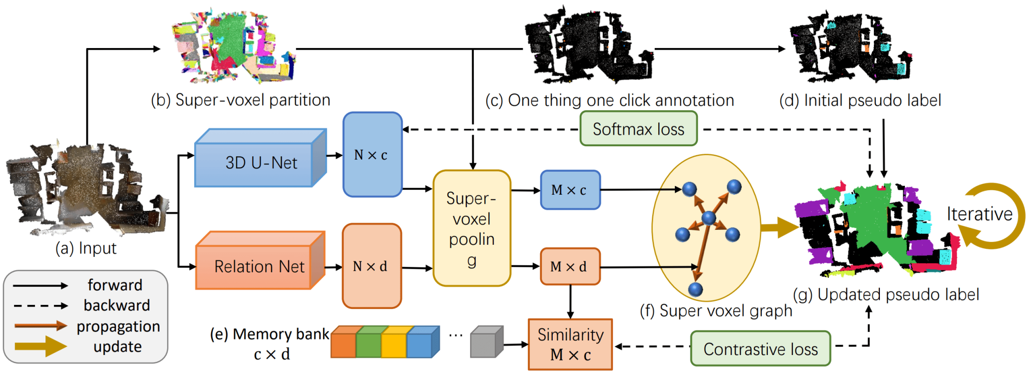


Figure 3.3: Overview of "One Thing One Click" framework, figure from [11].

methods are crucial. Researchers have proposed various approaches that leverage the strengths of both modalities to improve segmentation performance in the absence of fully labeled data.

One such approach is presented by [101], who propose a joint 2D-3D deep architecture for semantic point cloud segmentation. Their method utilizes a deep convolutional framework that is supervised by 2D annotations to segment 3D point clouds. To enforce correspondences between 2D and 3D mappings and address occlusions, they employ a novel re-projection method and an Observability Network (OBSNet). However, this method still relies on dense 2D ground truth labels for accurate 2D semantic segmentation.

[102] propose a network that combines sparse 2D bounding box labels with available 3D information. By exploiting both modalities, their method enhances the segmentation accuracy of 3D point clouds while utilizing the limited supervision provided by the bounding box labels.

In a 2D-3D joint framework for weakly supervised semantic segmentation, [103] leverages CAM [90] in both the 2D and 3D domains, bridging the gap between 2D pixels and 3D points through projection. They utilize the 2D CAM as self-supervision to improve the semantic perception of the 3D CAM, resulting in enhanced segmentation results.

While the aforementioned methods focus on indoor datasets, Superpixel-driven Lidar Representations (SLidR) [104] addresses the outdoor scenario. They propose a self-supervised 2D-to-3D representation distillation method, incorporating a superpixel to superpoint contrastive loss and a carefully designed image feature upsampling architecture. However, during training, the image branch of their network is frozen, which limits the joint training of the entire network. In contrast, [105] proposes a cross-modality framework that can be trained synchronously, effectively incorporating complementary information from unlabeled images. Their approach involves a dual-branch network and an active labeling strategy to maximize the potential of weak labels and achieve 2D-3D knowledge transfer.

In our robust label propagation approach, we also leverage the combined 2D-3D domains for weakly supervised semantic segmentation. By integrating both modalities, we enhance the robustness of the label propagation process, enabling more accurate and reliable segmentation results. The complementary nature of 2D appearance information and 3D geometric relations aids in improving the segmentation performance and addressing the challenges posed by noisy oversegmented point clouds and 2D images. Our approach benefits from the insights and methodologies proposed by the aforementioned studies, contributing to the advancement of robust weakly supervised semantic segmentation techniques.

These studies, including our own approach, highlight the effectiveness of combining 2D and 3D information for weakly supervised semantic segmentation. By leveraging the complementary strengths of both modalities, these approaches demonstrate improved segmentation performance and the ability to transfer knowledge between 2D and 3D domains. The integration of 2D and 3D data in weakly supervised scenarios enables the extraction of richer semantic representations and enhances the understanding of complex scenes, while also providing robustness to noisy oversegmented data.

3.3 Learning with Noisy Labels

The field of semantic segmentation in images and point clouds has witnessed significant advancements, enabling accurate object and boundary delineation. However, the performance of these models can be hindered by label noise, which arises from errors or inconsistencies in the ground truth annotations. Label noise poses a challenge to semantic segmentation as it can lead to overfitting and performance degradation. Therefore, developing effective strategies to learn from noisy labels is essential to enhance the robustness and generalization capabilities of semantic segmentation methods.

Various factors contribute to the presence of noisy labels, as discussed in section 2.5. Existing research in learning with noisy labels has primarily focused on fully supervised scenarios, where the label noise is often attributed to incorrect annotations. This is due to the inherent difficulty and high cost involved in obtaining ground truth labels with high accuracy. Although methods addressing label noise have predominantly targeted image classification and image segmentation tasks to improve noise tolerance, recent studies have recognized the need to address label noise specifically in the context of semantic segmentation.

To mitigate the impact of label noise, researchers have explored different approaches. One direction of research focuses on developing robust architectures and loss functions to handle noisy labels. For instance, [13] introduce a probabilistic graphical framework that incorporates latent variables to model the relationships between input images, class labels, and label noises for image classification. [14] provide sufficient conditions on loss functions that inherently tolerate label noise for multiclass classification problems. They demonstrate the robustness of mean absolute error (MAE) loss compared to commonly used categorical cross entropy (CCE) loss in the presence of label noise. [106] propose utilizing different losses for foreground-background and foreground-instance sub-tasks in instance segmentation. They employ the noise-robust loss, reverse cross entropy (RCE) loss [107] to prevent incorrect gradient guidance in the foreground-instance sub-task, while using the standard cross entropy (CE) loss to fully exploit correct gradient guidance in the foreground-background sub-task. COVID-19 Pneumonia Lesion segmentation network (COPLE-Net) [108] proposes a framework for learning from noisy labels in the context of pneumonia lesion segmentation from computed tomography (CT) scans of COVID-19 patients. They design a noise-robust Dice loss [46] and an adaptive self-ensembling approach, which involves an adaptive teacher and an adaptive student, to improve the performance in dealing with noisy labels.

In the domain of learning with noisy labels, a research direction that has gained attention involves loss adjustment techniques, which aim to mitigate the negative impact of label noise by adjusting the loss of training samples. These techniques can be further classified into several subcategories, namely loss correction, loss reweighting, label refurbishment, and meta-learning.

One method that falls under the category of loss correction is the Gold Loss Correction (GLC) method [109] which estimates a corruption matrix based on a model trained on clean samples, allowing for loss correction. Another approach [110] adopts a label refurbishment strategy specifically designed for semi-supervised semantic segmentation. Their framework employs CAM [90] to generate pixel-level labels for images that initially possess only image-level labels. By training a clean segmentation model with a small set of strong annotations and utilizing the CE loss, they differentiate between clean labels and noisy pixel-level labels. To correct noisy labels, they construct a superpixel-based graph that incorporates spatial adjacency and semantic similarity, propagating the clean labels using Graph Attention Network (GAT) [71]. The corrected pixel-level pseudo labels are then utilized to train a semantic segmentation model.

Moreover, the exploration of learning with noisy labels has extended to weakly supervised settings. [111] propose a novel approach to learning a semantic segmentation model from both weak and noisy labels, employing label refurbishment techniques. Their method involves oversegmenting each image into superpixels, propagating weak and potentially noisy image-level labels to the superpixel level, and subsequently correcting the noisy labels. By treating weakly supervised semantic segmentation as a noise reduction problem, they develop a superpixel label noise reduction model based on sparse learning with an efficient optimization algorithm.

Another subcategorized approach for handling noisy labels is loss reweighting. [15] focus on the problem of multiclass classification under label noise and investigate importance reweighting techniques [112]. Their method assigns smaller weights to falsely labeled data and greater weights to correctly labeled data, thereby adjusting the loss function accordingly. Active Bias [113] which assigns weights to uncertain examples with inconsistent label predictions based on their prediction variances during training. Another work, DualGraph [114] captures structural relations among labels using graph neural networks and reweights the samples according to the distribution relation, aiming to eliminate abnormal noise samples. [115] leverage meta-learning principles and propose an automatic reweighting algorithm that assigns weights to training examples based on their gradient directions.

In the context of point cloud semantic segmentation, the Point Noise-Adaptive Learning (PNAL) framework [16] tackles annotation noise by incorporating a point-level confidence selection mechanism and a label correction process at the cluster level. This framework aims to enhance the robustness of point cloud semantic segmentation models to noisy annotations while maintaining computational efficiency. An extension of PNAL, called PNAL-boundary [116], is proposed to correct labels near boundaries while preserving clean labels for inner points in instance-level label noise scenarios.

In our work, we extend the existing approaches and investigate learning with noisy labels in the context of weakly supervised semantic segmentation. We adapt loss adjustment strategies, such as loss reweighting, and utilize robust loss functions to develop more robust techniques for label propagation in oversegmented point clouds and 2D images with noisy labels. By addressing the challenges posed by label noise, our approach contributes to the development of more accurate and reliable weakly supervised semantic segmentation methods.

3.4 VCCS: Voxel Cloud Connectivity Segmentation

VCCS [28] is an unsupervised oversegmentation algorithm. This algorithm is specifically designed for point clouds and leverages voxel relationships and geometric features to generate supervoxels. The goal of VCCS is to produce meaningful segments that align with object boundaries in the observed 3D space. To achieve this, the algorithm employs a seeding methodology in 3D space and utilizes flow-constrained local iterative clustering, taking into account both color and geometric characteristics.

In our study, the VCCS algorithm plays a crucial role as a preprocessing step for point clouds. By applying VCCS, we generate supervoxels through oversegmentation, which serves as a fundamental component in our exploration of weakly supervised semantic segmentation with limited labeling resources. The generation of oversegmented point clouds is essential for employing label propagation techniques and obtaining additional supervised signals during the training process. Consequently, it is imperative to understand the underlying methodology of VCCS to address any noise or inaccuracies in object boundaries and develop appropriate solutions.

Preserving object boundaries is a critical aspect of the VCCS method. The algorithm relies on an adjacency graph to establish relationships between voxels, ensuring that supervoxels accurately capture object boundaries without crossing disconnected boundaries in 3D space. This adjacency graph facilitates the generation of supervoxels and aids in the subsequent segmentation process.

The process of supervoxel generation in VCCS involves the selection of initial seed points to initiate the segmentation process. The 3D space is divided into a voxelized grid using a specific resolution denoted as R_{seed}, and each occupied seeding voxel corresponds to a potential seed point. The seed point is determined by identifying the voxel in the point cloud that is closest to the center of the seeding voxel.

In the VCCS algorithm, distance calculation plays a vital role in the clustering process. To ensure efficient and effective clustering, the spatial component of distances is normalized based on the seed resolution R_{seed}. This normalization constrains the search space for each cluster to terminate at the neighboring cluster centers, promoting coherent supervoxel generation.

The distance measure used in VCCS is defined as follows:

$$D = \sqrt{\frac{\lambda D_c^2}{m^2} + \frac{\mu D_s^2}{3R_{seed}^2} + \epsilon D_{HiK}^2} \tag{3.1}$$

In this equation, λ, μ, and ϵ are parameters that control the influence of color, spatial distance, and geometric similarity, respectively, during the clustering process. D_s represents the spatial distance, which measures the proximity of points in 3D space. D_c represents the color distance, quantifying the dissimilarity in color attributes between points. Finally, D_{HiK} represents the distance in the Fast Point Feature Histograms (FPFH) space, which is calculated using the Histogram Intersection Kernel. This distance captures the similarity in local geometric properties and aids in distinguishing different regions within the point cloud.

By considering the combined effects of color, spatial proximity, and geometric similarity through the distance measure, VCCS achieves segmentation results by accurately delineating object boundaries and capturing meaningful geometric structures in the point cloud data.

Furthermore, VCCS involves the Flow Constrained Clustering stage, which is an iterative process that assigns voxels to supervoxels while considering connectivity and flow. This stage utilizes local k-means clustering to iteratively expand supervoxels and maintain spatial continuity within object boundaries. The iterative refinement in the flow constrained clustering algorithm enhances the accuracy and coherence of the supervoxel segmentation, aligning it more closely with the underlying geometric structures present in the point cloud data.

4 Methodology

4.1 Overview

This section provides an overview of the methodology employed to address the challenges associated with weakly supervised semantic segmentation in noisy oversegmented point clouds and 2D images. The objective is to enhance the accuracy and robustness of semantic segmentation under the sparse label setting, while leveraging the fusion of 2D and 3D modalities and exploring loss adjustment strategies.

In the context of sparse label settings, the methodology incorporates an oversegmentation approach to increase the supervision signal. This involves segmenting the point cloud into supervoxels, which group together points with similar characteristics. Additionally, superpixels are generated by projecting the supervoxels onto their corresponding 2D image counterparts, facilitating the integration of the 2D modality. The sparse labels obtained are then propagated to all points within the supervoxels and all pixels within the superpixels, effectively augmenting the number of labeled points/pixels.

However, oversegmentation introduces complications of its own. The supervoxels and superpixels generated may suffer from noise, as the employed oversegmentation method may not precisely preserve object boundaries. Consequently, the generated supervoxels and superpixels may contain points belonging to different objects or exhibit imprecise boundaries, posing challenges for accurate label propagation and subsequent semantic segmentation.

To address these challenges, our methodology investigates strategies to improve label propagation by leveraging the information obtained through oversegmentation while mitigating the challenges posed by oversegmentation noise, as illustrated in Figure 4.1. The goal is to enhance the accuracy and robustness of weakly supervised semantic segmentation in noisy oversegmented point clouds and 2D images.

The methodology encompasses several key techniques:

Adaptation of a Robust Architecture: By utilizing BPNet [85], we aim to develop a robust architecture that enhances label propagation. This network leverages both 3D geometric features and 2D complementary information, augmenting the network's resilience to noisy labels.

Label Propagation on Oversegmented Point Clouds and Images: Sparse labels

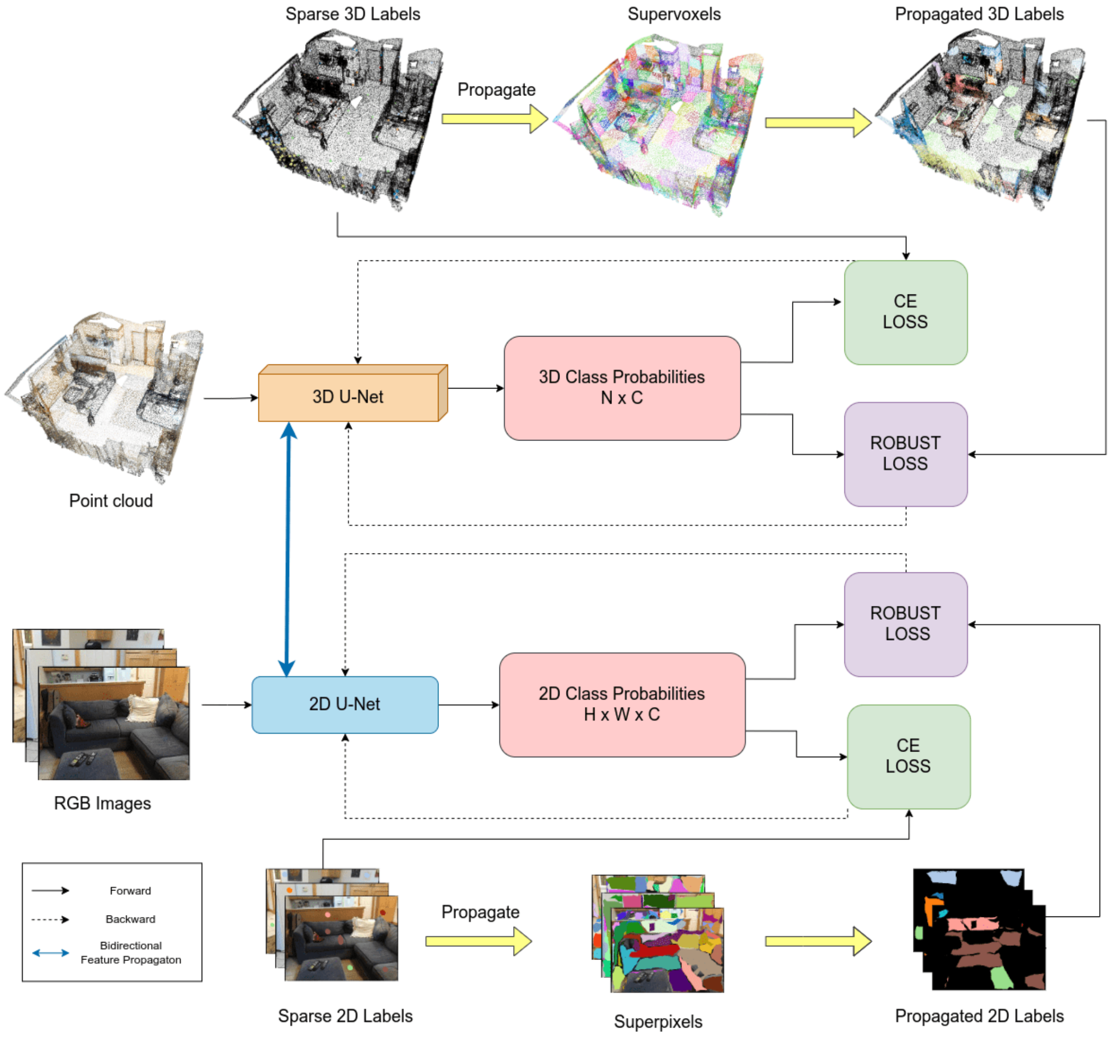


Figure 4.1: Overview of our framework. The framework employs supervoxel and superpixel partitioning to propagate sparse labels. We adopt BPNet [85] as the backbone architecture, which consist of two symmetric networks: 2D U-Net [20] and 3D MinkowskiUNet [21] for semantic label prediction. To mitigate label noise in the propagated labels, we incorporate robust loss functions and train the network using both CE loss and robust loss.

are propagated to unlabeled points/pixels within the oversegmented regions, thereby increasing the supervision signal and expanding the labeled dataset.

Loss Adjustment Methods by Loss Reweighting: To address the influence of noisy labels, loss adjustment techniques, such as loss reweighting, are investigated. These methods assign appropriate weights to training samples based on the distance metrics, reducing the impact of noisy labels during the training process.

Exploration of Robust Loss Functions: Various robust loss functions are experimented with to mitigate the negative impact of label noise. These loss functions aim to enhance the network's performance and improve its generalization capabilities.

By integrating these techniques, our methodology aims to develop more efficient and reliable methods for weakly supervised semantic segmentation, reducing the dependence on labor-intensive labeling processes and improving the generalization capabilities of deep learning models.

The subsequent sections will delve into the details of the data preprocessing stage, the approach for generating oversegmented point clouds and 2D images, the challenges associated with the employed oversegmentation method, and the robust techniques explored in this study, including the adaptation of a robust architecture, label propagation methods, the exploration of robust loss functions, and the generation of loss adjustment methods through loss reweighting.

4.2 Data Preprocessing

In this thesis, the data preprocessing stage involves preparing ScanNetV2 [5] and 2D-3D-S [17] for subsequent analysis and experimentation.

4.2.1 Generation of 2D Label Images

The generation of 2D label images is a crucial step in our methodology as they serve as input for our backbone network, which combines both 3D and 2D modalities.

To begin, we resized the color, depth, and pose images to a uniform resolution of 320 x 240 pixels. Subsequently, we employed perspective projection to create annotated label images. This process involved projecting the 3D points onto the 2D image plane and extracting the corresponding label information. This can be expressed as:

$$[u_i, v_i, 1]^{\mathrm{T}} = \mathcal{M}[x_i, y_i, z_i, 1]^{\mathrm{T}} \tag{4.1}$$

Here, $[x_i, y_i, z_i, 1]^{\mathrm{T}}$ represents the homogeneous coordinates of a 3D point in the world coordinate system, $[u_i, v_i, 1]^{\mathrm{T}}$ denotes its projected 2D homogeneous coordinates, and $\mathcal{M}$ is the perspective camera matrix derived from the intrinsic camera calibration matrix and the extrinsic camera pose matrix.

During the projection, several issues need to be addressed. To handle projected labels that reside outside the image boundary, we removed them from the resulting label images. Additionally, occlusion-related challenges were tackled by employing an occlusion mask during the projection process. This mask ensures that hidden surfaces, which are projected onto certain pixels but do not have an actual relationship with those pixels, are correctly handled. To achieve this, the depth map was utilized to determine if a point is occluded by comparing its depth value with the z-coordinate of the projected point located within a 5cm range from the 3D position of the corresponding pixel.

It is important to note that the generated label images may not possess the exact qualitative characteristics as the original image labels provided by the dataset, as additional filters and adjustments might be applied. However, for the purpose of our experiments and the goal of utilizing very sparse annotations in 3D, we generated our own 2D image labels to avoid an additional annotation burden for the 2D images. By generating our own 2D label images, we maintain control over the annotation process and can tailor it to the specific requirements of our methodology, while still achieving the objective of utilizing sparse annotations in the 3D domain.

4.2.2 Generation of Oversegmented Point Clouds

The generation of oversegmented point clouds plays a fundamental role in our study, specifically in the context of exploring weakly supervised semantic segmentation with very sparse labels. By leveraging label propagation, our aim is to obtain additional supervised signals during training, which necessitates the identification of meaningful regions within the point cloud.

To accomplish this, we employ the VCCS [28] algorithm as an unsupervised technique for generating oversegmented point clouds. The primary objective of the VCCS algorithm is to partition the point cloud data into semantically meaningful regions by utilizing spatial and normal characteristics. This process involves dividing the point cloud into smaller cubic regions called voxels using an octree structure. Subsequently, adjacent voxels are clustered using the k-means algorithm, resulting in the generation of supervoxels.

The calculation of distances plays a critical role in the VCCS algorithm as it determines the similarity between points. While the original VCCS paper incorporates color information in the distance calculation, we have decided to exclude color in our study due to its limited availability in various datasets. Instead, we have opted for a distance formula that prioritizes geometric similarity, as suggested by [23]:

$$D(p,q) = 1 - \left|n_p \cdot n_q\right| + 0.4\frac{|p-q|}{R_{\text{seed}}} \tag{4.2}$$

Here, n_p and n_q represent the normal vectors of points p and q, respectively. The distance formula combines the dot product of the normal vectors with the Euclidean distance between points, normalized by the parameter R_{seed}. By focusing exclusively on geometric similarity and excluding color information, our distance measure provides a robust foundation for the supervoxel segmentation process in our research.

However, a significant challenge associated with the VCCS algorithm is the presence of incorrect boundaries, particularly in cases where the point cloud exhibits non-uniform density. In such situations, multiple objects may overlap within the same voxel, resulting in points belonging to different objects existing within a supervoxel. This leads to supervoxels with inaccurate object boundaries, which can have a detrimental effect on the accuracy of assigned labels for each supervoxel. Consequently, the label propagation to each point within the supervoxel may suffer, potentially resulting in erroneous learning by the network.

To address this challenge, we incorporate robust techniques into our methodology to enhance the results of weakly supervised semantic segmentation. These techniques focus on mitigating the impact of noisy supervoxels and improving the accuracy of label assignment and subsequent label propagation. By employing these robust techniques, we aim to minimize the influence of incorrect boundaries and enhance the overall performance of weakly supervised semantic segmentation.

4.2.3 Assigning Labels to the Supervoxels

The process of assigning labels to the generated supervoxels is a crucial step in our methodology. This step aims to provide each supervoxel with a distinct and meaningful identity, which is essential for subsequent segmentation and analysis tasks.

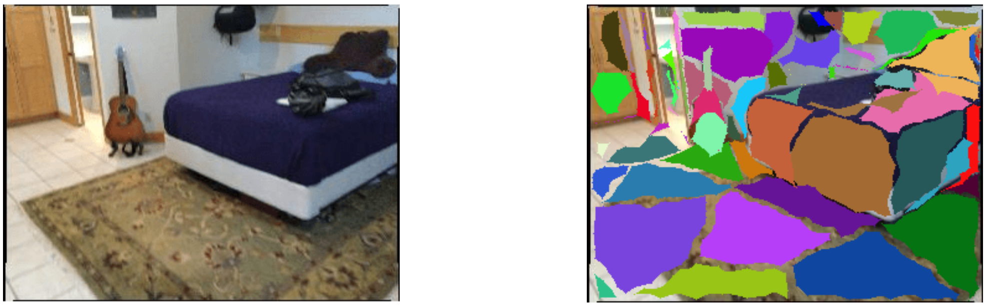
To assign labels to the supervoxels, we examine the points contained within each supervoxel. If a supervoxel does not contain any labeled points, no further action is taken, as it lacks sufficient information for reliable labeling. However, if the supervoxel comprises points that share a single label, we directly assign that label as the label for the entire supervoxel. This indicates that the supervoxel represents a coherent and homogeneous region with a clear semantic interpretation.

In certain cases, a supervoxel may contain points with multiple labels, indicating ambiguity or overlap between different object categories within the supervoxel. To address this ambiguity, we employ a majority voting scheme. We count the occurrences of each label within the supervoxel and assign the label with the highest count as the label for the supervoxel. This voting process ensures that the supervoxel is assigned a single label, even in the presence of multiple labels within it.

By assigning unique labels to the supervoxels, we establish an initial set of labels for each supervoxel. These initial labels serve as the starting point for the subsequent label propagation process.

4.2.4 Generation of Oversegmented Images

To generate oversegmented images, we employed perspective projection techniques by utilizing Equation 4.1. This process involved leveraging the color images, depth maps, intrinsic camera parameters, and pose information associated with the scene under consideration. By projecting the supervoxels onto the 2D image plane using the camera parameters and depth information, we were able to establish a mapping between the VCCS [28] supervoxels and the image space as illustrated in Figure 4.2.



2D Image

Our superpixels

Figure 4.2: Application of our superpixel generation algorithm on a 2D image from the ScanNetV2 dataset [5].

However, the generated superpixels may lack meaningful boundaries and connected components, which can hinder subsequent analysis and interpretation. To address this issue, we employed the concept of alpha shapes, also known as α-shapes, to refine the generated superpixels. Alpha shape computation provides a valuable tool for shape analysis by extending the concept of convex hulls to capture the interconnections between points within a finite set. It represents a family of piecewise linear curves in the Euclidean plane. The alpha parameter serves as a threshold value, defining the edges between points within a radius of $1/\alpha$ [117].

By incorporating the alpha shape computation method, we successfully generated oversegmented images with enhanced spatial continuity and coherence within object boundaries.

4.3 Label Propagation on Oversegmented Point Clouds and Images

Label propagation is a fundamental technique employed to enhance weakly supervised semantic segmentation by leveraging sparse labels. In this section, we outline the process of label propagation on oversegmented point clouds and images, aiming to enrich the training process and augment the limited supervision.

Once the supervoxels and superpixels are generated and assigned initial labels as described in the previous sections, the next step is to propagate these labels to the unlabeled points/pixels within the corresponding supervoxels and superpixels. By extending the labels, we obtain additional supervised signals, thereby enriching the training data and enhancing the discriminative capacity of the models. Figure 4.3 visually demonstrates the increase in the number of labeled points after label propagation.

However, it is crucial to acknowledge the challenges arising from inherent noise and the lack of well-defined object boundaries in oversegmented point clouds and images. These factors introduce potential inaccuracies during the label propagation process. For instance, the majority voting scheme employed in the initial label assignment may misclassify points/pixels belonging to different objects that reside within the same supervoxel or superpixel. Consequently, the entire region is assigned the label of the majority, resulting in misclassification.

The presence of noisy propagated labels poses challenges for deep learning models, as they are prone to biases originating from the training set. Therefore, it is essential to carefully consider and mitigate the effects of noisy labels during the training process. Strategies such as utilizing robust loss functions, regularization techniques, or loss adjustment methods can help alleviate the negative impact of noisy labels and enhance the robustness of the models.

Moreover, strategies such as incorporating confidence scores or weights for each label during the label propagation can help alleviate the negative impact of noisy labels and enhance the robustness of the models. By assigning higher confidence scores or weights to labels that exhibit greater consistency within the supervoxel or superpixel, the models can prioritize more reliable information and reduce the influence of noisy labels.

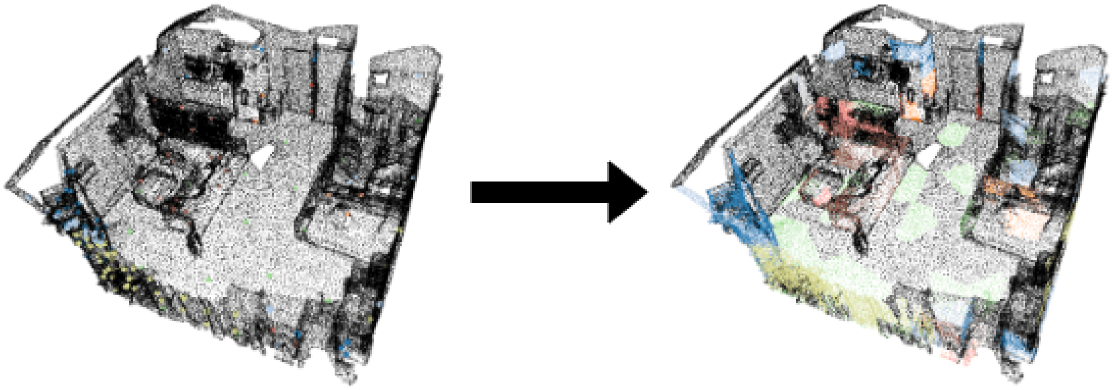


Figure 4.3: Illustration of label propagation on oversegmented point clouds. Sparse labels are propagated to unlabeled points within supervoxels, enriching the training data and increasing the number of labeled points.

4.4 Robust Architecture: Bidirectional Projection Network for Enhanced Weakly Supervised Semantic Segmentation

In our proposed framework, we adopt the BPNet [85] as the backbone network for semantic label prediction. The BPNet architecture is specifically designed to exploit the complementary information available in both the 2D and 3D domains, allowing for their seamless integration in an end-to-end fashion. Originally designed for fully supervised settings, BPNet has showcased remarkable performance on the ScanNetV2 [5] benchmark for both 2D and 3D semantic segmentation tasks. Notably, it has demonstrated the ability to distinguish geometrically close objects, such as walls and pictures, based on their 2D color and texture characteristics. Therefore, we adopt BPNet as the backbone network in our framework due to its characteristics aligning with our goal of developing a robust architecture within a weakly supervised setting.

To enhance the robustness of BPNet in a weakly supervised scenario, we propose an innovative approach by integrating additional components into the framework. These components play a pivotal role in guiding the training process by incorporating oversegmented point clouds and images. This integration leads to an increased number of labels obtained through label propagation, thereby enabling the network to gain a more comprehensive understanding of the scene and improving its semantic segmentation performance.

During the training phase, we utilize point clouds with sparse labels alongside their corresponding 2D images, which also possess limited annotations. These data pairs are fed as input to BPNet. The architecture of BPNet comprises two symmetric subnetworks: a 2D UNet [20], serving as the backbone for 2D segmentation, and a 3D MinkowskiUNet [21], serving as the backbone for 3D segmentation. The training loss employed by BPNet is a weighted sum of CE losses for the 2D and 3D predictions. In our framework, we further enhance the loss function by incorporating additional robust loss terms aimed at minimizing the discrepancy between the 2D and 3D predictions of BPNet and the oversegmented point clouds and pixels. This enhancement serves to align the predictions of BPNet with the additional supervisory signals, facilitating better consistency and improving the network's ability to handle the challenges posed by weak supervision and the inherent diversity of the input data.

4.5 Robust Losses for Handling Noisy Labeled Oversegmented Point Clouds and Pixels

In this section, we delve into the crucial aspect of handling noisy labeled oversegmented point clouds and images within our robust architecture. The presence of such noise poses a significant challenge in effectively training the adapted network, as it can lead to the learning of mislabeled data, ultimately impairing the overall performance of the network. To address this issue and ensure robustness, we explore the integration of additional components, specifically focusing on robust losses.

Robust losses play a vital role in reducing the network's sensitivity to noisy labels, enabling it to learn from the available data with greater resilience. By employing a robust loss instead of a non-robust one, the model becomes less influenced by the noisy labels and exhibits reduced sensitivity to large errors. Consequently, the robust loss functions serve as a means to establish the desired robustness in our network architecture.

The choice of a suitable loss or objective function holds paramount importance in the design of complex segmentation-based deep learning architectures. Researchers, recognizing this significance, have extensively investigated various domain-specific loss functions, aiming to enhance the results obtained on their respective datasets. In the subsequent sections, we present and elaborate on the specific loss functions that we have thoroughly investigated and experimented with in our approach.

By examining and assessing these robust loss functions, we aim to identify the most effective one for our methodology. The selected robust loss function will then be integrated into our framework, further enhancing its ability to handle noisy labeled oversegmented point clouds and pixels.

4.5.1 Cross Entropy Loss

The CE loss is a commonly used loss function in deep learning for classification tasks, such as semantic segmentation. It quantifies the dissimilarity between the predicted probability distribution and the true distribution of class labels. In the context of pixel-wise semantic segmentation, the network predicts the probability of each semantic category for each pixel in an image. The CE loss is computed as the average negative logarithm of the predicted probabilities for the correct classes.

The CE loss is defined as:

$$Loss_{CE} = -\frac{1}{\mathbf{N}} \sum_{n=1}^{\mathbf{N}} y_n \log \hat{y}_n \tag{4.3}$$

where **N** represent the total number of pixels in the image and y_n and $\hat{y}n$ denote the one-hot vector representation of ground truth labels and the corresponding softmax output from the network, respectively.

4.5.2 Dice Loss

Dice loss [46], is an objective function based on the Dice coefficient. The Dice coefficient measures the overlap between two sets and is commonly used for evaluating segmentation tasks. In the context of pixel-wise semantic segmentation, Dice loss addresses the issue of class imbalance in terms of pixel count between the foreground and background. In situations where foreground examples are extremely scarce in an image, the network may exhibit a strong bias toward the background. To address this problem, Dice loss is proposed to balance the foreground and background contributions. Dice loss is defined as:

$$Loss_{Dice} = 1 - \frac{2\sum_{i=1}^{n} p_i y_i}{\sum_{i=1}^{n} p_i^2 + \sum_{i=1}^{n} y_i^2} \tag{4.4}$$

where, $p_i \in P$ represents the predicted probability of the i-th pixel, and $y_i \in G$ represents the corresponding ground truth. The loss computes the Dice coefficient by calculating the overlap between the predicted and ground truth values and subtracting it from 1. By minimizing this loss, the network is encouraged to maximize the overlap between the predicted and ground truth segmentation maps, leading to improved segmentation performance.

4.5.3 Focal Loss

Focal loss [47], is a modification of the CE loss that addresses the issue of class imbalance by down-weighting the loss assigned to well-classified examples. This down-weighting mechanism enables the model to focus more on learning from hard examples, improving its performance in challenging scenarios. Focal loss aims to tackle the problem of overwhelming easy negatives during training by emphasizing a sparse set of hard examples. Focal loss is defined as:

$$Loss_{Focal} = -(1 - p_t)^{\gamma} \log(p_t) \tag{4.5}$$

In this formula, $(1 - p_t)$ acts as a modulating factor added to the CE loss, where p_t represents the predicted probability of the true class. The focusing parameter γ controls the degree of down-weighting. When an example is misclassified and p_t is small, the modulating factor is close to 1, and the loss remains unaffected. However, as p_t approaches 1, the factor approaches 0, effectively down-weighting the loss for well-classified examples. The focusing parameter γ allows for a smooth adjustment of the down-weighting rate for easy examples.

4.5.4 Tversky Loss

Tversky loss [48], is a loss function based on the Tversky index. It is designed to address the issue of data imbalance and improve segmentation outcomes with high precision but low recall. By incorporating weighting coefficients α and β, Tversky loss assigns different weights to false positives (FP) and false negatives (FN), allowing for a more flexible trade-off between precision and recall. Tversky loss is defined as:

$$Loss_{Tversky} = 1 - \frac{1 + p\hat{p}}{1 + p\hat{p} + \alpha(1 - p)\hat{p} + \beta p(1 - \hat{p})} \tag{4.6}$$

In this formula, p represents the predicted probability of a positive label, and $\hat{p}$ represents the ground truth probability. The numerator $1 + p\hat{p}$ measures the intersection between the predicted and ground truth labels, while the denominator incorporates additional terms that account for false positives and false negatives. The weighting coefficients α and β control the influence of FP and FN, respectively. Setting $\alpha = \beta = 0.5$ results in an equal weighting of precision and recall, equivalent to Dice loss. By adjusting the values of α and β, Tversky loss allows for prioritizing precision or recall, providing a flexible framework for handling imbalanced datasets and improving segmentation performance.

4.5.5 Focal Tversky Loss

Focal Tversky loss [118], is a modified version of the Tversky loss [48] function that addresses the issue of class imbalance in semantic segmentation tasks. It aims to improve the balance between precision and recall in the segmentation results. Similar to Focal loss [47], Focal Tversky loss incorporates a focal mechanism to emphasize hard examples, particularly those with small regions of interest (ROIs), by introducing a parameter γ. Focal Tversky loss is defined as:

$$Loss_{FocalTversky} = \sum_{c}(1 - TI_c)^{1/\gamma} \tag{4.7}$$

In this formula, c represents the class index, and TI_c denotes the Tversky index for class c. The Tversky index measures the similarity between the predicted and ground truth segmentations. By subtracting the Tversky index from 1, Focal Tversky loss assigns higher weights to examples with lower Tversky scores, which correspond to more challenging cases. The parameter γ controls the rate at which the loss is down-weighted for well-classified examples. Choosing γ within the range of [1, 3] allows for customization of the focal mechanism according to the desired emphasis on hard examples.

4.5.6 Online Hard Example Mining

The fundamental principle of Online Hard Example Mining (OHEM) [119] is to construct mini-batches using high-loss examples. This is accomplished by assigning a score to each training example based on its loss, reflecting the degree of difficulty encountered by the current network in classifying that specific example.

The procedure of OHEM can be described as follows: Given a list of training examples and their corresponding losses, the algorithm selects the example with the highest loss. Subsequently, any other examples with low training losses (typically those below 70% of the highest training loss) are discarded. This selection process is repeated until the desired batch size is attained, resulting in a mini-batch comprised of the highest-loss examples.

In contrast to the Focal loss [47], which assigns higher weights to misclassified examples while still considering easier examples, OHEM completely disregards easy examples during the training process. By solely focusing on challenging examples, OHEM aims to enhance the effectiveness and efficiency of training.

4.5.7 Lovasz-Softmax Loss

The Lovasz-Softmax loss [120], is a loss function specifically designed for optimizing the Mean Intersection over Union (mIoU) in neural networks for semantic segmentation tasks. It leverages the convex Lovasz extension of sub-modular losses to directly optimize the Intersection over Union (IoU).

The Lovasz-Softmax loss combines the softmax and Lovasz hinge functions to achieve tractable optimization and improved performance, particularly on small objects and false negatives. It operates on the normalized network outputs and is piecewise linear in these outputs. By considering the class-averaged mIoU metric common in semantic segmentation, the loss averages the class-specific surrogates. The Lovasz-Softmax loss is defined as:

$$Loss_{Lovasz-Softmax} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \overline{\Delta_{J_c}}(\boldsymbol{m}(c)) \tag{4.8}$$

Here, $\mathcal{C}$ represents the set of classes, and $|\mathcal{C}|$ denotes the number of classes. $\boldsymbol{m}(c)$ represents the network's predicted probability vector for class c. The function $\overline{\Delta_{J_c}}(\boldsymbol{m}(c))$ corresponds to the Lovasz hinge function, which quantifies the deviation between the predicted probabilities and the ground truth with respect to the Jaccard index.

In the case of confident outputs (large scores), where the probability vectors at each pixel closely resemble an indicator vector, the Lovasz-Softmax loss converges to the discrete Jaccard index for the corresponding discrete labeling with respect to the ground truth.

By directly optimizing the mIoU through the Lovasz-Softmax loss, the network is encouraged to produce more accurate and precise segmentations, especially for challenging cases involving small objects and false negatives. The convexity of the Lovasz extension enables efficient optimization, making the loss suitable for training neural networks in semantic segmentation tasks.

4.6 Loss Adjustment Methods for Weighted Impact and Improved Training

In this section, we present novel loss adjustment methods designed to handle noisy labeled oversegmented point clouds and pixels within our robust architecture. The aim is to effectively mitigate the negative impact of noisy labels by adjusting the loss of all training examples prior to updating the neural network weights. To achieve this, we propose loss reweighting, which assigns different confidence scores to each example, enabling a weighted training scheme.

By introducing an additional level of granularity into the segmentation process, we effectively address challenges such as noise, occlusion, and accurate delineation of complex object boundaries. These loss adjustment methods contribute to improved training by providing a more robust and reliable learning framework.

In the following subsections, we will describe the specific algorithms used to calculate the distances and weighting algorithms for confidence score assignment, which serve as key components of our loss adjustment strategy.

4.6.1 Distance Calculation Algorithms

To implement loss adjustment, we have developed distance calculation algorithms that take into account the geometric relationships within the clusters. In the context of this section, clusters refer to supervoxels in point clouds and superpixels in images. These algorithms enable the calculation of distances between the points/pixels within each cluster and the known sparse labels provided by the dataset, as well as the cluster centers. By considering the geometric relationships within the clusters, we aim to assign a proper confidence score that takes into account the presence of noisy boundaries containing mislabeled points.

The distance calculation algorithms are applied separately for supervoxels in point clouds and superpixels in images. The calculated distances play a crucial role in creating the weights for loss adjustment.

4.6.1.1 Center-Based Distance Calculation

The center-based distance calculation algorithm is used to compute the distances between the points or pixels within a cluster and the center of that cluster. The main idea behind this algorithm is that propagated labels of points or pixels closer to the cluster center are considered more reliable. By assessing the distances between the labels and their respective cluster centers, we can evaluate the reliability of the labels.

Algorithm 1 Center-Based Distance Calculation

```
Input: Point cloud data or image data with propagated labels
Output: Distances between propagated labels and cluster centers
for each cluster cl in clusters do
    Compute the center $\mathbf{c}$ = calculateCenter($\mathbf{cl}$)
    for each point/pixel p in cl do
        Calculate the distance $\mathbf{d}$ = calculateDistance($\mathbf{p}, \mathbf{c}$)
    end for
end for
```

The Algorithm 1 utilizes the `calculateCenter` function to determine the center of each cluster. Then, for each point or pixel within the cluster, the algorithm calculates the distance between the label and its corresponding cluster center using the `calculateDistance` function. This process is repeated for all clusters in the point cloud or image data. The resulting distances provide valuable information about the reliability of the propagated labels based on their proximity to the cluster centers. By considering the distances from the cluster centers, the algorithm assigns higher reliability to the propagated labels of points or pixels that are closer to the center of their respective clusters.

4.6.1.2 Closest Sparse Label to Center-Based Distance Calculation

The closest sparse label to center-based distance calculation method focuses on identifying the closest sparse label to the center of the cluster. The algorithm calculates the distances between the sparse label and all the points or pixels with the propagated label within the cluster. The assumption is that the sparse label closest to the cluster center provides more reliable information than other sparse labels in the cluster.

Algorithm 2 Closest Sparse Label to Center-Based Distance Calculation

```
Input: Point cloud data or image data with propagated labels, Sparse labels
Output: Distances between closest sparse label and propagated labels
for each cluster $\mathbf{cl}$ in clusters do
    Compute the center $\mathbf{c} = \text{calculateCenter}(\mathbf{cl})$
    Determine the closest sparse label $\mathbf{csl} = \text{min}(\text{calculateDistance}(\mathbf{c}, \text{sparseLabels}))$
    for each point/pixel $\mathbf{p}$ in $\mathbf{cl}$ do
        Calculate the distance $\mathbf{d} = \text{calculateDistance}(\mathbf{p}, \mathbf{csl})$
    end for
end for
```

The Algorithm 2 iterates over each cluster in the given point cloud or image data with propagated labels. For each cluster, it calculates the center using the `calculateCenter` function. It then determines the closest sparse label to the center by finding the minimum distance between the center and the sparse labels. Next, for each point or pixel within the cluster, it calculates the distance between the point and the closest sparse label using the `calculateDistance` function. This process is repeated for all clusters, resulting in the distances between the closest sparse label and the propagated labels. This method allows for the identification of the most relevant sparse label within each cluster based on its proximity to the cluster center.

4.6.1.3 Multiple Sparse Label-Based Distance Calculation

The multiple sparse label-based distance calculation algorithm aims to quantify the distances between the sparse labels and the points/pixels with propagated labels within the clusters. By considering all sparse labels within each cluster, the algorithm provides a comprehensive analysis of the distances between points.

This approach acknowledges the spatial relationships between points/pixels and assigns higher reliability to propagated labels that are in close proximity to the sparse labels. By considering multiple sparse labels within each cluster, the algorithm provides valuable guidance for accurate labeling.

Algorithm 3 Multiple Sparse Label-Based Distance Calculation

```
Input: Point cloud data or image data with propagated labels, Sparse labels
Output: Distances between sparse labels and propagated labels
for each cluster cl in clusters do
    for each point/pixel p in cl do
        Initialize the distance $\mathbf{d} = 0$
        for each sparseLabel s in cl do
            Increment the distance $\mathbf{d} = \mathbf{d} + \text{calculateDistance}(\mathbf{s}, \mathbf{p})$
        end for
        Calculate the average distance $\mathbf{d} = \mathbf{d}/\text{count}(\mathbf{s})$
    end for
end for
```

The Algorithm 3 iterates over each supervoxel in the point cloud and each superpixel in the image. For each supervoxel or superpixel, it calculates the distance between each sparse labels and the points/pixels within the corresponding cluster. The distances are then averaged by dividing the cumulative distance by the number of sparse labels in the cluster. This ensures that the resulting distances provide a measure of proximity between the sparse labels and the propagated labels within the cluster.

4.6.1.4 Center Weighted Multiple Sparse Label-Based Distance Calculation

The Center Weighted Multiple Sparse Label-Based Distance Calculation algorithm is designed to calculate the distances between the sparse labels and the points with propagated labels within the cluster with an emphasis on weighting the sparse labels based on the distance to the cluster center. This method combines the principles of Center-Based Distance Calculation in 4.6.1.1 and Multiple Sparse Label-Based Distance Calculation in 4.6.1.3. Additionally, it introduces a weighting mechanism based on the distance of each sparse label to the cluster center. The objective is to assign higher importance to sparse labels that are closer to the cluster center. The algorithm follows these steps:

The Algorithm 4 starts by initializing the center of each cluster as the average coordinates of the points or pixels within the cluster. It calculates the distance of the sparse label to the cluster center and computes the distance between the point or pixel and the sparse label. Additionally, it calculates the weight of the sparse label based on its distance to the cluster center. The weight is determined by the ratio of the distance

Algorithm 4 Center Weighted Multiple Sparse Label-Based Distance Calculation

```
Input: Point cloud data or image data with propagated labels, Sparse labels
Output: Distances between propagated labels and sparse labels, Sparse label weights
for each cluster cl in clusters do
    Compute the center $\mathbf{c}$ = calculateCenter($\mathbf{cl}$)
    Compute the distance fd of furthest point/pixel in cl
    for each point/pixel p in cl do
        Initialize the total distance $\mathbf{d} = 0$
        for each sparse label s in cl do
            Calculate the distance of s to c, $\mathbf{sc}$ = calculateDistance($\mathbf{s}, \mathbf{c}$)
            Increment the distance $\mathbf{d} = \mathbf{d}$ + calculateDistance($\mathbf{s}, \mathbf{p}$)
            Calculate the weight of s, $\mathbf{ws} = 1 - (\mathbf{sc}/\mathbf{fd})$
        end for
        Calculate the average distance $\mathbf{d} = \mathbf{d}$/count($\mathbf{s}$)
    end for
end for
```

between the sparse label and the cluster center to the distance of the furthest point or pixel in the cluster to the cluster center. Finally, the total distance is normalized by dividing it by the number of sparse labels in the cluster. This normalization ensures a consistent interpretation of the distances across different supervoxels or superpixels.

The sparse label weights obtained from this algorithm will be used to scale the weight calculation described in Section 4.6.2. By incorporating the weight mechanism based on the proximity to the cluster center, the algorithm assigns greater influence to sparse labels that are closer to the center. This approach enhances the accuracy and reliability of the previous multiple sparse label-based distance calculation method.

4.6.2 Weighting Algorithms for Confidence Score Assignment

Based on the calculated distances, we employ weighting algorithms to assign confidence scores to individual points or pixels. Inspired from [112], we assign smaller confidence scores to examples with greater distances and greater confidence scores to those with smaller distances. The assigned confidence scores reflect the likelihood of points having the correct label, facilitating a more informed loss adjustment.

The update equation for the network parameters is given as:

$$\Theta_{t+1} = \Theta_t - \eta \nabla \Big(\frac{1}{|\mathcal{B}t|} \sum (x, \tilde{y}) \in \mathcal{B}_t w(x, \tilde{y}) \ell\big(f(x; \Theta_t), \tilde{y}\big) \Big), \tag{4.9}$$

where $\mathcal{B}_t$ represents a mini-batch of training examples at iteration t, x is an example from the mini-batch, $\bar{y}$ is the corresponding noisy label, and $\ell\big(f(x;\Theta_t),\bar{y}\big)$ denotes the loss function that measures the discrepancy between the predicted output $f(x;\Theta_t)$ and the noisy label $\bar{y}$. The weight $w(x,\bar{y})$ is assigned to each example x and its corresponding noisy label $\bar{y}$ based on the confidence score assigned to that example.

Based on the distances calculated in section 4.6.1, we introduce three different weighting algorithms to assign confidence scores to individual points or pixels: linear weighting, power weighting, and Gaussian weighting.

4.6.2.1 Linear Weighting

The linear weighting algorithm calculates the weights based on the distances between points within the cluster using the following equation:

$$w(x,\bar{y}) = 1 - \frac{\text{distance}}{\max(\text{distances})} \tag{4.10}$$

4.6.2.2 Power Weighting

The power weighting algorithm assigns weights based on a power function applied to the distances. It calculates the weights using the following equation:

$$w(x,\bar{y}) = e^{-p\cdot\text{distance}} \tag{4.11}$$

where p denotes the power factor of the weighting.

4.6.2.3 Gaussian Weighting

The Gaussian weighting algorithm assigns weights based on a Gaussian distribution of the distances. It calculates the weights using the following equation:

$$w(x,\bar{y}) = e^{-\text{distance}} \tag{4.12}$$

These weighting algorithms enable the assignment of confidence scores to individual points based on their distances within the cluster. The resulting weights can be used in the update equation for the network parameters, as described earlier, to adjust the confidence scores and facilitate informed loss adjustment.

5 Experiments and Results

This chapter presents the experimental evaluation of the proposed methodology on various datasets. It includes comparisons of different methods, analysis of parameter variations, and the identification of the best-performing combinations.

5.1 Datasets

This study performs experiments on two prominent semantic segmentation datasets: ScanNetV2 [5] and 2D-3D-S [17]. These datasets are carefully selected to showcase the proposed methodology and facilitate comparisons with other existing methods. ScanNetV2 and 2D-3D-S are widely acknowledged benchmarks in the field of 3D real-world semantic segmentation.

5.1.1 ScanNetV2

ScanNetV2 [5] represents an RGB-D video dataset consisting of 2.5 million views captured from 1513 distinct scenes. Each scene is meticulously annotated with 3D camera poses, surface reconstructions, and semantic segmentations. The dataset encompasses a wide range of indoor environments, including kitchens, dining rooms, and bedrooms, with 20 semantic labels provided for each scene. A comprehensive annotation effort resulted in 3391 annotation tasks performed on the 1513 scans, which were subsequently split into 1201 training and 312 validation scans. The richness of annotations and the diverse nature of ScanNetV2 have positioned it as a widely utilized benchmark in the field of semantic segmentation.

During our experiments, we preprocess the 3D data by extracting point clouds and voxelizing them into 3D volumes. For the 2D input, we adopt the approach described in section 4.2.1 to avoid the additional burden of annotating 2D images. However, it is important to acknowledge that our baseline architecture, BPNet [85], employs label images from the ScanNetV2 dataset. These RGB images have undergone additional filtering and adjustments, resulting in higher quality and less noisy 2D inputs. This preprocessing step may impact the performance of our model by providing a more refined and enhanced 2D input representation.

5.1.2 2D-3D-S

The 2D-3D-S dataset [17] represents a collection of large-scale indoor spaces, consisting of 271 rooms with 13 distinct object categories. This dataset serves as an extension of the S3DIS dataset [18] and was captured from six expansive indoor areas spanning three different buildings. Within the 2D-3D-S dataset, various data modalities are provided, including RGB, depth, and global XYZ OpenEXR images, along with corresponding 3D meshes and point clouds for each indoor space. Notably, the 3D point cloud data within the 2D-3D-S dataset has also been utilized in the S3DIS dataset.

We specifically chose the 2D-3D-S dataset for our experiments due to the availability of RGB images. This dataset offers a crucial ingredient for generating such labels, namely the global XYZ files. These files contain ground truth locations of each image pixel in the mesh and are stored as 16-bit, 3-channel OpenEXR files. By leveraging these global XYZ files, we were able to generate accurate 2D image labels from the corresponding 3D point clouds. To ensure a consistent evaluation, we adhered to the official train/validation split provided by the dataset. Our annotation efforts focused on Areas 1, 2, 3, 4, and 6 of the 2D-3D-S dataset, while the performance evaluation was conducted on Area 5.

5.2 Annotation Details

In this study, we employed various annotation strategies to establish weakly supervised semantic segmentation settings and obtain training signals with varying levels of supervision.

1 Labeled Point per Supervoxel strategy: For each computed supervoxel, we randomly select a point and seek a semantic label from the corresponding 3D annotated point cloud. We made the assumption that semantic annotations within each supervoxel are clean and consistent, allowing us to consider 1 labeled point in a supervoxel as equivalent to labeling all points within that supervoxel. This strategy enables us to compare our results with fully supervised 3D semantic segmentation networks, even though the supervoxel boundaries may be noisy and the supervoxel labels may have some inaccuracies. After label expansion, we created a dense setting for evaluation, facilitating a fair comparison with fully supervised approaches.

Moreover, the "1 labeled point per supervoxel" strategy corresponds to 0.2% supervision for both the ScanNetV2 [5] and 2D-3D-S [17] datasets.

ScanNetV2 Data Efficient Benchmark [8]: This benchmark is also employed to compare our method with other studies. It considers four different training configurations on ScanNetV2, including using 20, 50, 100, and 200 labeled points per scene. In our experiments, we report our results on the most challenging setting with only 20 annotated points. With 20 labeled points per scene, an annotator only needs to label the semantic labels for 20 points, significantly reducing the time and cost compared to fully annotating all points in a room.

Random Point Selection: This approach is established to obtain 0.02% supervision for both the ScanNetV2 and 2D-3D-S datasets. We sampled 0.02% of the points as annotated points from the original 3D point cloud and left the rest of the points unlabeled. Additionally, for the 2D-3D-S dataset, we randomly sampled 20 points from the original point cloud for our experiments.

5.3 Evaluation Metric

In evaluating the performance of our algorithms on each dataset, we utilize the mIoU as the primary evaluation metric. The mIoU measures the average IoU for each class, providing an overall assessment of the segmentation accuracy.

The IoU, also known as the Jaccard index, quantifies the similarity between two sets by dividing the size of their intersection by the size of their union. It is defined as:

$$IoU = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{5.1}$$

Here, TP represents the true positives, FP represents the false positives, and FN represents the false negatives. In the context of semantic segmentation, TP refers to the number of correctly predicted foreground pixels, FP refers to the number of background pixels predicted as foreground, and FN refers to the number of foreground pixels missed by the model.

To evaluate the segmentation performance of each class, we calculate the IoU score between the predicted mask and the ground-truth mask for that specific class. If there are K classes in total, the mIoU is computed as the average of the IoU scores for all classes:

$$\mathrm{mIoU} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{IoU}_k \tag{5.2}$$

The mIoU serves as a measure of the overall segmentation performance, where a higher mIoU score indicates better accuracy in distinguishing foreground classes from the background class. By evaluating the pixelwise similarity between the ground-truth and predicted segmentation masks, the mIoU metric provides a comprehensive assessment of the algorithm's performance or error in semantic segmentation.

5.4 Implementation Details

The proposed framework is implemented using PyTorch [121], PyTorch Lightning [122], and the MinkowskiEngine sparse convolution library [21]. The implementation is trained on two NVIDIA A40 GPUs for a total of 100 epochs.

During training, we set the mini-batch size, base learning rate, momentum, and weight decay to 16, 0.01, 0.9, and 0.0001, respectively. The stochastic gradient descent (SGD) [123] optimizer is utilized, and we employ a polynomial learning rate scheduler with a power of 0.9 to adaptively adjust the learning rate.

For our final model, we select Dice loss [46] as the robust loss function and set the supervoxel resolution to 0.4 meters using VCCS [28]. We employ the center weighted multiple sparse label-based distance calculation algorithm with Gaussian weighting for confidence score assignment.

For the training process, we use the ScanNetV2 [5] and the 2D-3D-S dataset [17]. The 2D UNet [20] component of our framework is initialized with weights pretrained on ImageNet [124], while the 3D component is initialized from scratch.

The final training objective combines all the individual objectives related to the 2D and 3D predictions. It can be expressed as:

$$L_{total} = \lambda_{\mathrm{3D}}(\lambda_{\mathrm{CE_{3D}}} L_{\mathrm{CE_{3D}}} + \lambda_{\mathrm{Robust_{3D}}} L_{\mathrm{Robust_{3D}}}) + \lambda_{\mathrm{2D}}(\lambda_{\mathrm{CE_{2D}}} L_{\mathrm{CE_{2D}}} + \lambda_{\mathrm{Robust_{2D}}} L_{\mathrm{Robust_{2D}}}) \tag{5.3}$$

Here, $L_{\mathrm{CE_{3D}}}$ represent the cross-entropy loss for the 3D sparse labels and $L_{\mathrm{Robust_{3D}}}$ represent the robust loss for the propagated supervoxel labels, respectively. Similarly, $L_{\mathrm{CE_{2D}}}$ denote the cross-entropy loss for 2D sparse labels and $L_{\mathrm{Robust_{2D}}}$ denotes the robust loss for the propagated superpixel labels. The parameters $\lambda_{\mathrm{CE_{3D}}}$, $\lambda_{\mathrm{Robust_{3D}}}$,$\lambda_{\mathrm{CE_{2D}}}$, $\lambda_{\mathrm{Robust_{2D}}}$ and, λ_{3D} are empirically set to 1, while λ_{2D} is set to 0.1 in our experiments.

5.5 Results

In this section, we will compare our method with existing methods for ScanNetV2 [5], and 2D-3D-S [17] datasets.

5.5.1 Evaluations on ScanNetV2

Our framework is evaluated on the ScanNetV2 dataset [5] using three weakly-supervised settings: "1 labeled point per supervoxel," 20 points annotated per scene, and 0.02% of points annotated in each scene.

Model	Supervision	mIoU (%)
PointNet++ [2]	100%	33.9
SPLATNet [125]	100%	39.3
TangentConv [126]	100%	43.8
PointCNN [4]	100%	45.8
3DMV [82]	100%	48.4
FPConv [63]	100%	63.9
PointConv [62]	100%	66.6
KPConv [61]	100%	68.4
MinkowskiNet [21]	100%	73.6
BPNet [85]	100%	**74.9**
Ours	1 labeled point per supervoxel	67.0

Table 5.1: Quantitative results mIoU(%) of existing methods on the ScanNetV2 [5] online test set.

We first present the performance of our framework on the ScanNetV2 online test set. In this evaluation, we compare our proposed approach, trained under the "1 labeled point per supervoxel" setting, with several state-of-the-art fully supervised methods for 3D semantic segmentation. The results of this comparison are summarized in Table 5.1. Notably, despite the sparse annotation of only 0.2% of the points, our framework achieves superior performance, surpassing many existing fully supervised approaches.

Next, we analyze the results on the ScanNetV2 validation set, as presented in Table 5.2. In the "1 labeled point per supervoxel" setting, our framework is compared against the fully supervised MinkowskiNet [21] method. Notably, our framework demonstrates a 1.0% higher mIoU compared to MinkowskiNet. This improvement can be attributed to the robustness of our backbone network, which effectively leverages both 2D and 3D features. Although our results closely align with those of BPNet [85], the slight performance gap can be attributed to the noisy boundaries generated by our oversegmentation strategy for supervoxels and superpixels. Additionally, when comparing our framework to Supervoxel-CNN [84] and BPNet†, we achieve higher mIoU values by 3.3% and 0.2%, respectively. These results highlight the competitive performance of our framework even when not all points are annotated.

In the 20 points setting, as defined by the ScanNetV2 Efficient Benchmark [8], our framework outperforms both BPNet† and OTOC [11]. Specifically, we achieve a 2.43% higher mIoU compared to BPNet† and a 2.26% higher mIoU compared to OTOC. This demonstrates the effectiveness of our framework in utilizing the limited supervision of 20 annotated points per scene.

Model	Supervision	mIoU (%)
MinkowskiNet [21]	100%	68.0
BPNet [85]	100%	**70.6**
BPNet [85]†	1 labeled point per supervoxel	68.8
Supervoxel-CNN [84]	1 labeled point per supervoxel	65.7
Ours	1 labeled point per supervoxel	**69.0**
OTOC [11]*	20 points	55.06
BPNet [85]†	20 points	54.89
Ours	20 points	**57.32**
OTOC [11]*	0.02%	**62.18**
BPNet [85]†	0.02%	57.86
Ours	0.02%	59.20

Table 5.2: mIoU(%) of our results and baselines with diverse supervision on the ScanNetV2 dataset [5] validation set. † means the BPNet [85] model trained with labels propagated on oversegmented point clouds and images. * means OTOC [11] baseline model trained with the initial pseudo labels.

In the 0.02% points setting, our framework surpasses BPNet† by a 1.34% increase in mIoU, showcasing the benefits of incorporating robust components to enhance generalization and performance. However, it is important to note that our framework is outperformed by OTOC, which achieves a higher mIoU of 2.98%. This performance difference can be attributed to the utilization of the provided segments in OTOC as supervoxels, which benefit from a denser coverage in the ScanNetV2 dataset.

To ensure a fair comparison, we consider the methodology employed by OTOC, as discussed in [9]. OTOC utilizes the provided segments in ScanNetV2 as the basis for supervoxel partitioning, resulting in pure and consistent labels for the points within each supervoxel after oversegmentation. Additionally, OTOC determines its labeling ratio by calculating the number of clicks divided by the total number of raw points. This approach assumes clean and consistent semantic annotations within each supervoxel, where a single click per supervoxel is equivalent to labeling all points within that supervoxel. Consequently, the supervoxel semantic labels utilized by OTOC exhibit denser coverage in the ScanNetV2 dataset, surpassing the annotation ratio of 0.02%.

In contrast, our approach in the 0.02% point per scene setting follows the methodology of prior works such as [6] and [7], utilizing the total number of labeled points for evaluation.

5.5.2 Evaluations on 2D-3D-S

To further validate the effectiveness of our proposed framework, we conducted evaluations on the 2D-3D-S dataset [17]. We considered two weakly supervised settings: "1 labeled point per supervoxel" and 0.02% supervision. The results of our framework in these settings are presented in Table 5.3.

In the "1 labeled point per supervoxel" setting, where only a single labeled point was assigned to each supervoxel (equivalent to 0.2% annotated points), our framework outperformed several existing fully supervised 3D semantic segmentation networks. Specifically, our framework surpassed the baseline method BPNet† [85] by 0.52% mIoU.

In the 0.02% supervision setting, our framework demonstrated remarkable performance. We outperformed the state-of-the-art OTOC [11] by a substantial margin of 17.02% mIoU. Additionally, our framework surpassed the baseline method BPNet† by 0.08% in the same setting. These results highlight the effectiveness of our proposed framework in achieving accurate semantic segmentation even with extremely limited supervision. The evaluation results on the 2D-3D-S dataset provide strong evidence of the superior performance of our framework compared to existing approaches. Our framework exhibits robustness and adaptability to different levels of annotation sparsity.

Model	**Supervision**	**mIoU (%)**
PointNet [2]	100%	41.1
SegCloud [3]	100%	48.9
TangentConv [126]	100%	52.8
PointCNN [4]	100%	57.3
SuperpointGraph [100]	100%	58.0
KPConv [61]	100%	**67.1**
MinkowskiNet [21]	100%	65.4
BPNet [85]†	1 labeled point per supervoxel	64.76
Ours	1 labeled point per supervoxel	**65.28**
OTOC [11]*	%0.02	43.07
BPNet [85]†	%0.02	60.01
Ours	%0.02	**60.09**

Table 5.3: Quantitative results mIoU(%) of existing methods and baselines with diverse supervision on the 2D-3D-S [17] Area-5. † means the BPNet [85] model trained with labels propagated on oversegmented point clouds and images. * means OTOC [11] baseline model trained with the initial pseudo labels.

5.6 Qualitative Results

In addition to the quantitative evaluation, we present qualitative results of our framework on both the ScanNetV2 [5] and 2D-3D-S [17] datasets, as shown in Figure 5.1 and Figure 5.2, respectively. Our framework was trained under the "1 labeled point per supervoxel" supervision and compared against our baseline BPNet [85] trained under full supervision. The superior segmentation accuracy of our proposed framework is highlighted by the red bounding boxes in the figures.

Furthermore, our qualitative results demonstrate the robustness of our framework in handling noisy labels and accurately delineating object boundaries. The segmentation outputs exhibit clearer and more precise object boundaries compared to the baseline method. Additionally, our framework successfully captures fine-grained details and textures in complex scenes, leading to improved semantic segmentation results.

These qualitative results provide visual evidence of the superiority of our proposed framework in achieving accurate and detailed semantic segmentation results, even with limited supervision. The visual comparisons against the baseline method validate the effectiveness and potential of our approach in real-world applications.

5.7 Ablation Studies

To assess the effectiveness of individual modules and analyze the impact of various design choices on the model's performance, we conduct extensive ablation studies. These studies are conducted with the purpose of gaining deeper insights into the inner workings of our framework and conducting a comprehensive evaluation of its constituent components. The evaluation process involves using the ScanNetV2 [5] validation dataset, while the training phase incorporates the ScanNetV2 data efficient benchmark [8], which includes annotations of 20 points per scene. The results of each ablation study are presented in terms of mIoU.

5.7.1 Effectiveness of combining 2D and 3D features

To evaluate the effectiveness of incorporating both 2D and 3D features, we conduct a detailed comparative analysis between two variants of our model. The first variant employs 2D-3D fusion using the BPNet [85] backbone, while the second variant relies solely on 3D features and is trained with the MinkowskiNet [21] backbone.

The results presented in Table 5.4 clearly demonstrate the performance advantage gained by incorporating additional 2D feature information. This advantage can be attributed to the inherent limitations of 3D data, which often lacks fine texture details

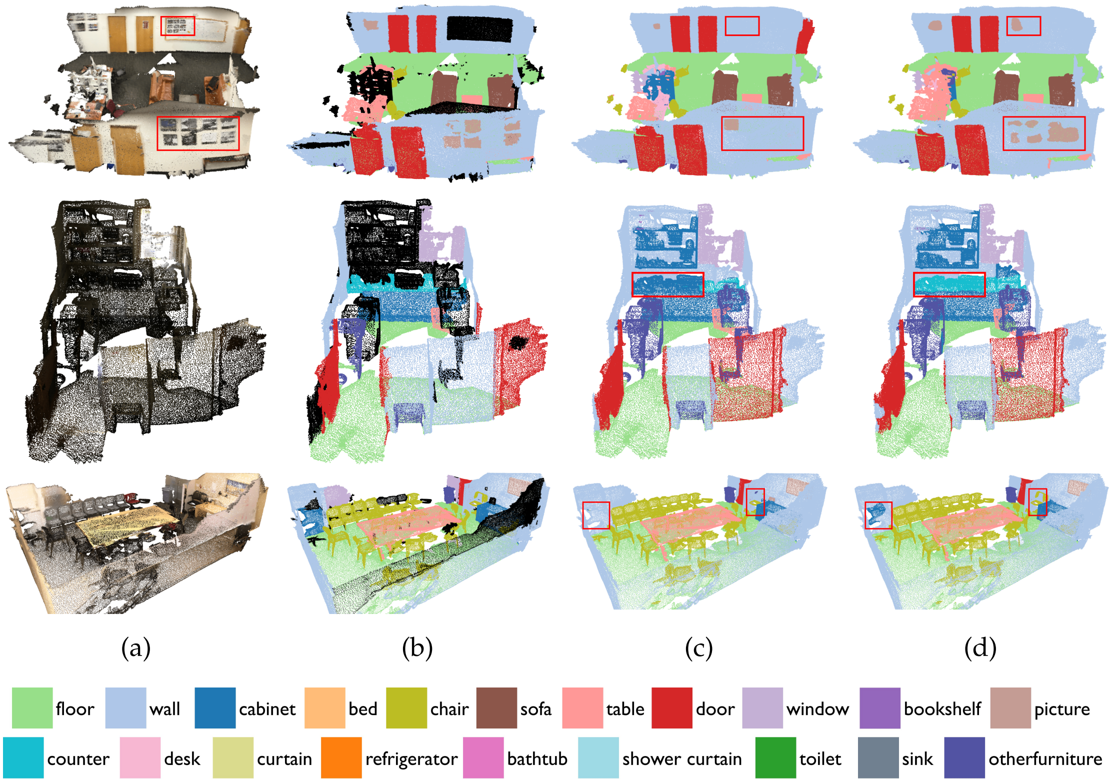


Figure 5.1: Qualitative results on ScanNetV2 dataset [5] on validation set. (a) Input point cloud, (b) Ground truth, (c) BPNet [85] with 100% supervision, (d) Ours with "1 labeled point per supervoxel" annotation setting. We highlight the differences between the results of our method and fully supervised baseline by red boxes.

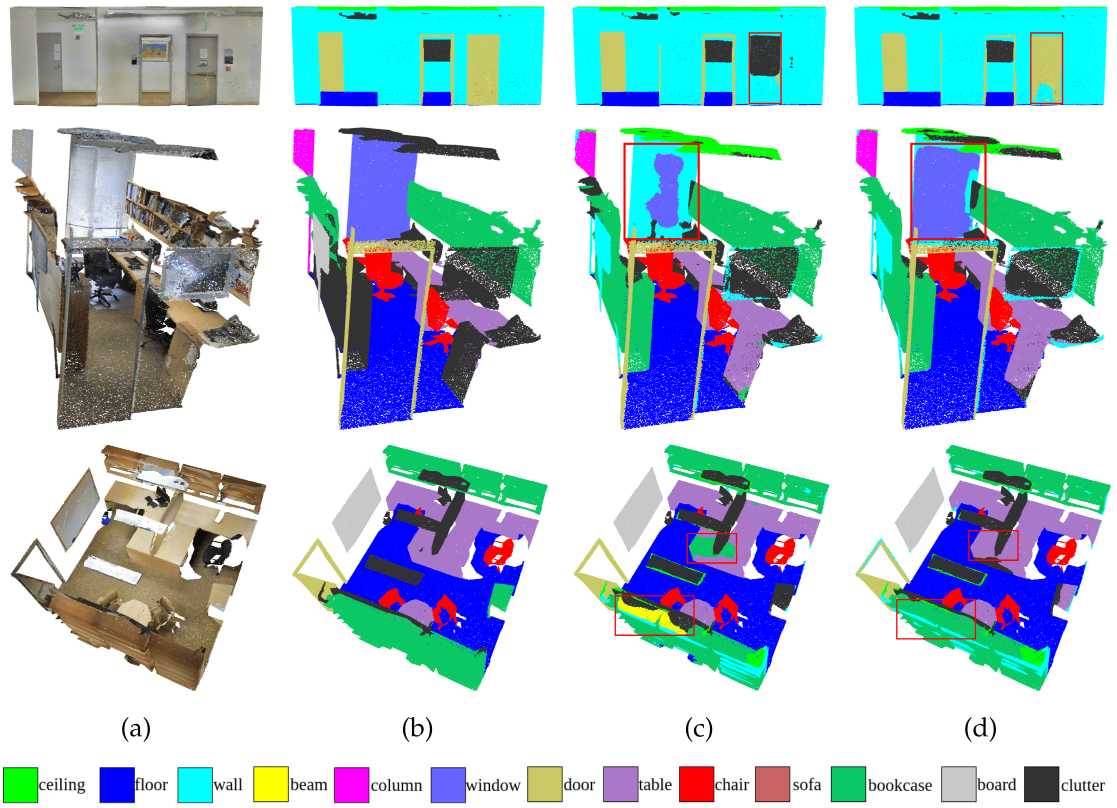


Figure 5.2: Qualitative results on 2D-3D-S [17] Area-5. (a) Input point cloud, (b) Ground truth, (c) BPNet [85] with 100% supervision, (d) Ours with "1 labeled point per supervoxel" annotation setting. We highlight the differences between the results of our method and fully supervised baseline by red boxes.

necessary for accurate semantic predictions. By integrating high-quality texture information extracted from 2D images, our proposed method improves the accuracy and robustness of semantic predictions, showcasing the effectiveness of utilizing 2D features to enhance the understanding of 3D models.

Model	**mIoU (%)**
Ours (2D + 3D)	**56.23**
Ours (3D-only)	53.68

Table 5.4: Comparison of effectiveness of combining 2D and 3D features.

5.7.2 Comparison of Different Robust Losses

In our framework, the selection of a suitable robust loss function is crucial for effectively handling noisy labels. A robust loss helps improve the generalization capability of our framework, even in the presence of label noise. Hence, we conduct a comprehensive analysis to determine the most appropriate robust loss for our system.

As presented in Table 5.5, among the evaluated robust losses, the Dice loss [46] outperforms the others, achieving a mIoU score of 55.92%. By optimizing the dice loss, our framework demonstrates improved performance in accurately segmenting objects with noisy labels.

Robust Losses	**mIoU (%)**
Cross Entropy Loss	54.55
Focal Loss [47]	54.67
Dice Loss [46]	**55.92**
LovaszSoftmax Loss [120]	55.82
Ohem Loss [119]	55.4
Tversky Loss [48]	54.77
Focal Tversky Loss [118]	55.12

Table 5.5: Comparison of different robust loss functions.

Based on the results, we conclude that the Dice loss is the most suitable robust loss function for our framework. By addressing class imbalance and emphasizing precise segmentation boundaries, the Dice loss enhances the accuracy of semantic segmentation, particularly in the presence of noisy labels.

5.7.3 Effect of Varying Supervoxel Resolution

It is important to note that the supervoxels generated by VCCS [28] may not accurately preserve object boundaries. Hence, it is crucial to investigate the impact of different supervoxel resolutions on the performance of our framework. Finding the optimal supervoxel resolution that strikes a balance between preserving object boundaries and capturing smaller structures is of utmost importance.

To evaluate the effect of varying supervoxel resolutions on our framework's performance, we conduct comprehensive ablation studies using different resolutions, as illustrated in Figure 5.3. The objective of these ablations is to identify the supervoxel resolution that yields the best results in terms of preserving object boundaries. The outcomes of these experiments are summarized in Table 5.6, where it can be observed that the supervoxel resolution of 0.4 meters achieved the highest mIoU score among the tested resolutions.

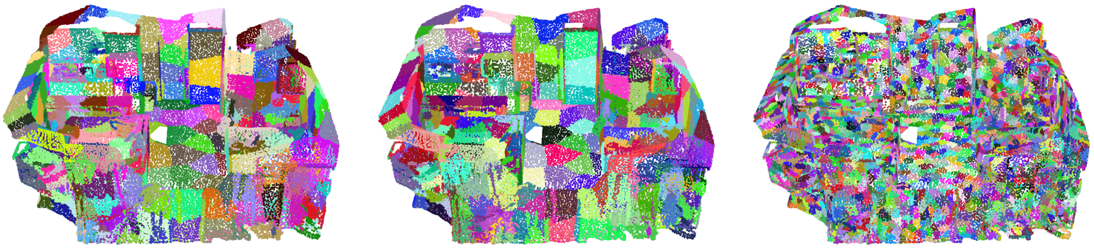
Supervoxel Resolution (meters)	**mIoU (%)**
0.6	56.17
0.4	**56.68**
0.2	55.92

Table 5.6: Comparison of different supervoxel resolutions.

Analyzing the results, we observe that the use of a 0.6 meter supervoxel resolution results in a slightly lower mIoU score compared to 0.4 meters. This decrease in performance can be attributed to the larger size of supervoxels at 0.6 meters. The larger supervoxels may merge smaller structures, potentially leading to inaccuracies in preserving object boundaries.

Similarly, the employment of a 0.2 meter supervoxel resolution also exhibits a lower mIoU score compared to 0.4 meters. This can be attributed to the smaller size of supervoxels at 0.2 meters, which may accurately capture finer details and boundaries. However, the smaller supervoxels also increase the likelihood of merging adjacent structures into separate supervoxels. Consequently, this can result in fragmented object representations and a decrease in overall performance.

The results from our ablation studies emphasize the significance of selecting an appropriate supervoxel resolution that strikes a balance between preserving object boundaries and avoiding the merging of adjacent structures. Based on the experimental findings, a supervoxel resolution of 0.4 meters demonstrates superior performance, making it the most suitable choice for our framework.



(a) 0.6 Voxel size

(b) 0.4 Voxel size

(c) 0.2 Voxel size

Figure 5.3: Visual comparison of varying supervoxel resolutions.

5.7.4 Impact of Distance Calculation Algorithms

In this section, we investigate the influence of different distance calculation algorithms on determining the confidence of each point and pixel within a supervoxel and superpixel. We explore various approaches to quantify the distance and measure their effect on performance.

Table 5.7 presents the results of these strategies and highlights the performance of each distance calculation algorithm. Among the tested approaches, the algorithm that considers the distance of points and pixels to the center and annotated label outperforms the others in terms of mIoU score.

Distance Calculation Algorithms	**mIoU (%)**
Center-Based	55.76
Closest Sparse Label to Center-Based	55.92
Multiple Sparse Label-Based	55.89
Center Weighted Multiple Sparse Label-Based	**56.4**

Table 5.7: Comparison of different distance calculation algorithms.

This outcome suggests that the supervoxels and superpixels derived from oversegmented point clouds and pixels often exhibit noisy boundaries. By assigning a higher confidence to points that are closer to the center and annotated label, our proposed approach accounts for this noise and improves the accuracy of the corresponding points.

Analyzing the specific algorithms, we observe that the center-based distance calculation achieves a mIoU score of 55.76%. Although this approach provides a reasonable measure, it does not fully capture the information regarding the annotated labels.

The multiple sparse label-based distance calculation achieves a slightly higher mIoU score of 55.89%. This approach considers the distance between the annotated label and the propagated labels, providing a more comprehensive representation of the supervoxel or superpixel.

Further enhancement is achieved by employing the center weighted multiple sparse label-based distance calculation, which achieves a mIoU score of 56.4%. This algorithm assigns a higher weight to the annotated label and accounts for the proximity of points and pixels to the center.

Lastly, the closest sparse label to center-based distance calculation yields a mIoU score of 55.92%. Although this algorithm considers the proximity of the cluster center to the closest annotated label, it does not fully capture the confidence variation within the supervoxel or superpixel.

The importance of being close to the center in oversegmentation is highlighted by our findings, as the distance calculation algorithm that incorporates the distance to the center and annotated label achieves the highest mIoU score. This emphasizes the significance of points and pixels that are near the representative center, allowing for improved confidence estimation within the supervoxel or superpixel. By considering proximity to the center, our algorithm effectively addresses noisy boundaries, resulting in enhanced performance in 3D semantic segmentation.

5.7.5 Impact of Weighting Algorithms

In this section, we explore different weighting algorithms to assign weights to points within supervoxels and pixels within superpixels based on their calculated distance values obtained from the distance calculation algorithm. For power weighting, we set the power factor, p, to 10. The results are shown in Table 5.8.

Weighting Algorithms	**mIoU (%)**
Linear	55.36
Power	55.24
Gaussian	**55.92**

Table 5.8: Comparison of different weighting algorithms.

Among the tested algorithms, the Gaussian weighting algorithm demonstrates superior performance, achieving a mIoU score of 55.92%. This algorithm considers the probabilistic distribution of distances and assigns higher weights to points or pixels with smaller distances, resulting in improved confidence estimation within the segments.

The linear weighting algorithm, which assigns weights linearly based on distance values, achieves a mIoU score of 55.36%. While providing a basic weighting mechanism, it fails to capture confidence variations accurately.

Similarly, the power weighting algorithm, which applies a power function to distance values, achieves a slightly lower mIoU score of 55.24%. Although it attempts to emphasize points or pixels with smaller distances, it tends to be excessively strict in its weighting scheme.

Based on the results, we conclude that the Gaussian weighting algorithm outperforms the linear and power weighting algorithms. By considering the probabilistic distribution of distances, it effectively assigns higher weights to points and pixels with smaller distance metrics, leading to improved confidence estimation and more accurate semantic segmentation.

5.7.6 Effect of Oversegmentation Strategy

The boundaries of the generated supervoxels from VCCS [28] may contain noise, which can result in noisy labels when propagating the labels. In order to evaluate the impact of noisy boundaries, we compare the results of our framework using supervoxels generated by VCCS with the ground truth segments provided by the ScanNetV2 [5].

Table 5.9 presents the comparison between our framework's performance using the ground truth supervoxels and the supervoxels generated by VCCS. Surprisingly, despite the presence of noisy boundaries in the supervoxels generated by VCCS, our framework achieved superior results compared to using the ground truth supervoxels.

Oversegmentation Strategy	**mIoU (%)**
Ours w/ supervoxels provided by ScanNetV2	56.16
Ours w/ supervoxels generated by VCCS	**56.23**

Table 5.9: Comparison of the effect of oversegmentation strategy.

This suggests that our robust learning framework compensates for the noise introduced by the oversegmentation step. VCCS captures meaningful structures within the point clouds by leveraging local geometric cues, resulting in a reasonable partitioning of the data. Although the boundaries may contain noise, the overall partitioning aligns well with the underlying structures and semantic regions. Additionally, our framework incorporates robust learning techniques, such as the robust loss functions, which help mitigate the negative impact of the noisy boundaries and encourage the model to focus on informative cues within the supervoxels.

6 Conclusion and Future Work

In this thesis, we address the issue of label noise in weakly supervised semantic segmentation, with a focus on oversegmented point clouds and images with imprecise object boundaries. The primary objective is to improve label propagation in these noisy environments, aiming to enhance the robustness of segmentation results.

Recognizing the limitations of fully supervised semantic segmentation, which heavily relies on extensively annotated datasets, the potential of weakly supervised learning techniques is explored. We propose a framework that integrates data from both 2D and 3D domains, leveraging the rich texture and geometric information available. The supervisory signals are strengthened by grouping points and pixels with similar attributes, achieved by oversegmenting point clouds and images. Initial labels are assigned to supervoxels and superpixels using a novel label assignment strategy, and these labels are propagated to unlabeled points or pixels within the corresponding regions.

Despite the innovative approach, our framework faces challenges due to imprecise object boundaries in the oversegmented regions. This leads to inaccuracies in the propagated labels and the emergence of label noise. To address this, we propose a novel noise-robust framework that focuses on enhancing the network's learning capacity and enabling robust learning with limited annotations. By incorporating limited supervision, the annotation overhead is reduced, leading to the development of more efficient methods for robust point cloud semantic segmentation.

A key contribution of this thesis is the development of a robust framework, which effectively handles label noise in oversegmented point clouds and images. The introduction of novel loss adjustment strategies enables the network to cope with noisy labels during the training process. Through the assignment of appropriate weights based on the distance metrics and the incorporation of multi-modality, our framework exhibits increased resilience against label noise, leading to more reliable and accurate segmentation results.

Moreover, the integration of robust loss into the framework contributes to improved learning ability and generalizability. By experimenting with different domain-specific robust loss functions and selecting the most suitable one, our framework achieves better alignment with ground truth annotations and improved segmentation accuracy.

Experiments conducted on two large 3D datasets, ScanNetV2 [5], and 2D-3D-S [17], demonstrate the superiority of the proposed approach, achieving remarkable results in 3D semantic segmentation with extremely sparse annotations and outperforming the baselines by a significant margin. However, the proposed methodology faces certain limitations. The generation of 2D labels directly from sparse point cloud data poses challenges, leading to lower-quality 2D labels compared to the annotated 2D labels in the dataset. Additionally, the inherent noise in the generated superpixels and occlusions results in superpixels without any sparse labels within their regions, adversely affecting distance calculations for these superpixels.

Our research offers several promising research directions for future work. One potential direction involves exploring different oversegmentation algorithms. Adopting a differentiable oversegmentation methodology that allows backpropagation can improve the definition of object boundaries. Another potential area for improvement is the initial label assignment process. Incorporating all labels within oversegmented regions, instead of relying solely on majority voting, can capture more nuanced semantic information, enhancing the precision and robustness of semantic segmentation.

Abbreviations

VCCS Voxel Cloud Connectivity Segmentation

SLIC Simple Linear Iterative Clustering

SSN Superpixel Sampling Network

BESS Boundary-Enhanced Supervoxel Segmentation

SSP Supervized SuperPoint

GAN generative adversarial network

RCGAN Robust Conditional GAN

EM expectation-maximization

FCNs Fully Convolutional Networks

MLPs Multi-Layer Perceptrons

LiDAR Light Detection and Ranging

MRI Magnetic Resonance Imaging

COO Coordinate list

CNNs convolutional neural networks

CRF conditional random field

A-SCN Attentional ShapeContextNet

PATs Point Attention Transformers

GAC Graph Attention Convolution

SSCN Submanifold Sparse Convolutional Networks

D-CNN Depth-aware CNN

MVPNet Multi-View PointNet

BPNet Bidirectional Projection Network

BPM bidirectional projection module

CAM classification activation maps

SEC Seed, Expand, and Constrain

SPN Superpixel Pooling Network

PSD Perturbed self-distillation

WyPR Weakly-supervised framework for Point cloud Recognition

GaIA Graphical information gain-based attention network

OTOC One Thing One Click

OBSNet Observability Network

SLidR Superpixel-driven Lidar Representations

MAE mean absolute error

CCE categorical cross entropy

CE cross entropy

RCE reverse cross entropy

CT computed tomography

COPLE-Net COVID-19 Pneumonia Lesion segmentation network

GAT Graph Attention Network

GLC Gold Loss Correction

PNAL Point Noise-Adaptive Learning

FPFH Fast Point Feature Histograms

FP false positives

FN false negatives

ROIs regions of interest

OHEM Online Hard Example Mining

IoU Intersection over Union

mIoU Mean Intersection over Union

List of Figures

List of Tables

Bibliography

[1] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. "Matterport3d: Learning from rgb-d data in indoor environments." In: *arXiv preprint arXiv:1709.06158* (2017).

[2] 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.

[3] 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.

[4] 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).

[5] 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.

[6] X. Xu and G. H. Lee. "Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 13706–13715.

[7] Y. Zhang, Z. Li, Y. Xie, Y. Qu, C. Li, and T. Mei. "Weakly supervised semantic segmentation for large-scale point cloud." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 4. 2021, pp. 3421–3429.

[8] J. Hou, B. Graham, M. Nießner, and S. Xie. "Exploring data-efficient 3d scene understanding with contrastive scene contexts." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15587–15597.

[9] 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: *European Conference on Computer Vision*. Springer. 2022, pp. 600–619.

[10] J. Wei, G. Lin, K.-H. Yap, T.-Y. Hung, and L. Xie. "Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 4384–4393.

[11] 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.

[12] Y. Wu, S. Cai, Z. Yan, G. Li, Y. Yu, X. Han, and S. Cui. "PointMatch: A Consistency Training Framework for Weakly Supervised Semantic Segmentation of 3D Point Clouds." In: *arXiv preprint arXiv:2202.10705* (2022).

[13] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. "Learning From Massive Noisy Labeled Data for Image Classification." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[14] A. Ghosh, H. Kumar, and P. S. Sastry. "Robust loss functions under label noise for deep neural networks." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1. 2017.

[15] R. Wang, T. Liu, and D. Tao. "Multiclass learning with partially corrupted labels." In: *IEEE transactions on neural networks and learning systems* 29.6 (2017), pp. 2568–2580.

[16] S. Ye, D. Chen, S. Han, and J. Liao. "Learning with noisy labels for robust point cloud segmentation." In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 6443–6452.

[17] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. "Joint 2d-3d-semantic data for indoor scene understanding." In: *arXiv preprint arXiv:1702.01105* (2017).

[18] 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 conference on computer vision and pattern recognition*. 2016, pp. 1534–1543.

[19] 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.

[20] 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.

[21] 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*. 2019, pp. 3075–3084.

[22] 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.

[23] Y. Lin, C. Wang, D. Zhai, W. Li, and J. Li. "Toward better boundary preserved supervoxel segmentation for 3D point clouds." In: *ISPRS journal of photogrammetry and remote sensing* 143 (2018), pp. 39–47.

[24] Ren and Malik. "Learning a classification model for segmentation." In: *Proceedings ninth IEEE international conference on computer vision*. IEEE. 2003, pp. 10–17.

[25] 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.

[26] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz. "Superpixel sampling networks." In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 352–368.

[27] C. Xu and J. J. Corso. "Evaluation of super-voxel methods for early video processing." In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 1202–1209.

[28] 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*. 2013, pp. 2027–2034.

[29] S. Song, H. Lee, and S. Jo. "Boundary-enhanced supervoxel segmentation for sparse outdoor LiDAR data." In: *Electronics Letters* 50.25 (2014), pp. 1917–1919.

[30] 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.

[31] M. IEEE Standards Coordinating Committee et al. "IEEE standard glossary of software engineering terminology (IEEE Std 610.12-1990). Los Alamitos." In: *CA: IEEE Computer Society* 169 (1990), p. 132.

[32] N. Drenkow, N. Sani, I. Shpitser, and M. Unberath. "A systematic review of robustness in deep learning for computer vision: Mind the gap?" In: *arXiv preprint arXiv:2112.00639* (2021).

[33] T. Dreossi, S. Ghosh, A. Sangiovanni-Vincentelli, and S. A. Seshia. "A formalization of robustness for deep neural networks." In: *arXiv preprint arXiv:1903.10033* (2019).

[34] N. Nigam, T. Dutta, and H. P. Gupta. "Impact of noisy labels in learning techniques: a survey." In: *Advances in Data and Information Sciences: Proceedings of ICDIS 2019*. Springer. 2020, pp. 403–411.

[35] D. Karimi, H. Dou, S. K. Warfield, and A. Gholipour. "Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis." In: *Medical image analysis* 65 (2020), p. 101759.

[36] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee. "Learning from noisy labels with deep neural networks: A survey." In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[37] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. "Training convolutional networks with noisy labels." In: *arXiv preprint arXiv:1406.2080* (2014).

[38] K. K. Thekumparampil, A. Khetan, Z. Lin, and S. Oh. "Robustness of conditional gans to noisy labels." In: *Advances in neural information processing systems* 31 (2018).

[39] J. Goldberger and E. Ben-Reuven. "Training deep neural-networks using a noise adaptation layer." In: *International conference on learning representations*. 2016.

[40] A. Hernández-Garcıa and P. König. "Data augmentation instead of explicit regularization." In: *arXiv preprint arXiv:1806.03852* (2018).

[41] A. Krogh and J. Hertz. "A simple weight decay can improve generalization." In: *Advances in neural information processing systems* 4 (1991).

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[43] C. Shorten and T. M. Khoshgoftaar. "A survey on image data augmentation for deep learning." In: *Journal of big data* 6.1 (2019), pp. 1–48.

[44] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. "Understanding deep learning (still) requires rethinking generalization." In: *Communications of the ACM* 64.3 (2021), pp. 107–115.

[45] A. Krizhevsky, G. Hinton, et al. "Learning multiple layers of features from tiny images." In: (2009).

[46] 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.

[47] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. "Focal loss for dense object detection." In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[48] S. S. M. Salehi, D. Erdogmus, and A. Gholipour. "Tversky loss function for image segmentation using 3D fully convolutional deep networks." In: *International workshop on machine learning in medical imaging*. Springer. 2017, pp. 379–387.

[49] K. O'Shea and R. Nash. "An introduction to convolutional neural networks." In: *arXiv preprint arXiv:1511.08458* (2015).

[50] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks." In: *Advances in neural information processing systems* 25 (2012).

[51] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." In: *arXiv preprint arXiv:1409.1556* (2014).

[52] 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.

[53] 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.

[54] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "Semantic image segmentation with deep convolutional nets and fully connected crfs." In: *arXiv preprint arXiv:1412.7062* (2014).

[55] F. Yu and V. Koltun. "Multi-scale context aggregation by dilated convolutions." In: *arXiv preprint arXiv:1511.07122* (2015).

[56] W. Liu, A. Rabinovich, and A. C. Berg. "Parsenet: Looking wider to see better." In: *arXiv preprint arXiv:1506.04579* (2015).

[57] 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.

[58] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. "Context encoding for semantic segmentation." In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 7151–7160.

[59] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. "Ccnet: Criss-cross attention for semantic segmentation." In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 603–612.

[60] 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).

[61] 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*. 2019, pp. 6411–6420.

[62] 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.

[63] 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.

[64] 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)*. 2018, pp. 87–102.

[65] 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.

[66] 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.

[67] 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*. 2020, pp. 11108–11117.

[68] S. Xie, S. Liu, Z. Chen, and Z. Tu. "Attentional shapecontextnet for point cloud recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4606–4615.

[69] W. Zhang and C. Xiao. "PCAN: 3D attention map learning using contextual information for point cloud based retrieval." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12436–12445.

[70] J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian. "Modeling point clouds with self-attention and gumbel subset sampling." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3323–3332.

[71] 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*. 2019, pp. 10296–10305.

[72] 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*. 2019, pp. 10433–10441.

[73] 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.

[74] 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.

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

[76] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. "Volumetric and multi-view cnns for object classification on 3d data." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5648–5656.

[77] E. E. Aksoy, S. Baci, and S. Cavdar. "Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving." In: *2020 IEEE intelligent vehicles symposium (IV)*. IEEE. 2020, pp. 926–932.

[78] 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*. 2018, pp. 9224–9232.

[79] 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*. 2020, pp. 2940–2949.

[80] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. "3d graph neural networks for rgbd semantic segmentation." In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5199–5208.

[81] W. Wang and U. Neumann. “Depth-aware cnn for rgb-d segmentation.” In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 135–150.

[82] 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)*. 2018, pp. 452–468.

[83] 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 Workshops*. 2019, pp. 0–0.

[84] S.-S. Huang, Z.-Y. Ma, T.-J. Mu, H. Fu, and S.-M. Hu. “Supervoxel convolution for online 3d semantic segmentation.” In: *ACM Transactions on Graphics (TOG)* 40.3 (2021), pp. 1–15.

[85] 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*. 2021, pp. 14373–14382.

[86] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft coco: Common objects in context.” In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[87] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. “Fully convolutional multi-class multiple instance learning.” In: *arXiv preprint arXiv:1412.7144* (2014).

[88] A. Kolesnikov and C. H. Lampert. “Seed, expand and constrain: Three principles for weakly-supervised image segmentation.” In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer. 2016, pp. 695–711.

[89] S. Kwak, S. Hong, and B. Han. “Weakly supervised semantic segmentation using superpixel pooling network.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.

[90] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. “Learning deep features for discriminative localization.” In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.

[91] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. “Object region mining with adversarial erasing: A simple classification to semantic segmentation approach.” In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1568–1576.

[92] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang. "Weakly-supervised semantic segmentation network with deep seeded region growing." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7014–7023.

[93] J. Ahn and S. Kwak. "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4981–4990.

[94] J. Dai, K. He, and J. Sun. "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation." In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1635–1643.

[95] M. Rajchl, M. C. Lee, O. Oktay, K. Kamnitsas, J. Passerat-Palmbach, W. Bai, M. Damodaram, M. A. Rutherford, J. V. Hajnal, B. Kainz, et al. "Deepcut: Object segmentation from bounding box annotations using convolutional neural networks." In: *IEEE transactions on medical imaging* 36.2 (2016), pp. 674–683.

[96] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3159–3167.

[97] Y. Zhang, Y. Qu, Y. Xie, Z. Li, S. Zheng, and C. Li. "Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation." In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 15520–15528.

[98] Z. Ren, I. Misra, A. G. Schwing, and R. Girdhar. "3d spatial recognition without spatially labeled 3d." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13204–13213.

[99] M. S. Lee, S. W. Yang, and S. W. Han. "Gaia: Graphical information gain based attention network for weakly supervised point cloud semantic segmentation." In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 582–591.

[100] 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.

[101] H. Wang, X. Rong, L. Yang, J. Feng, J. Xiao, and Y. Tian. *Weakly Supervised Semantic Segmentation in 3D Graph-Structured Point Clouds of Wild Scenes*. 2020. arXiv: 2004.12498 [cs.CV].

[102] W. Sun, J. Zhang, and N. Barnes. "3D Guided Weakly Supervised Semantic Segmentation." In: *Proceedings of the Asian Conference on Computer Vision (ACCV)*. Nov. 2020.

[103] H. Kweon and K.-J. Yoon. "Joint Learning of 2D-3D Weakly Supervised Semantic Segmentation." In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 30499–30511.

[104] 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 (CVPR)*. June 2022, pp. 9891–9901.

[105] T. Sun, Z. Zhang, X. Tan, Y. Qu, Y. Xie, and L. Ma. *Image Understands Point Cloud: Weakly Supervised 3D Semantic Segmentation via Association Learning*. 2022. arXiv: `2209.07774 [cs.CV]`.

[106] L. Yang, F. Meng, H. Li, Q. Wu, and Q. Cheng. "Learning with noisy class labels for instance segmentation." In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer. 2020, pp. 38–53.

[107] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey. "Symmetric cross entropy for robust learning with noisy labels." In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 322–330.

[108] G. Wang, X. Liu, C. Li, Z. Xu, J. Ruan, H. Zhu, T. Meng, K. Li, N. Huang, and S. Zhang. "A noise-robust framework for automatic segmentation of COVID-19 pneumonia lesions from CT images." In: *IEEE Transactions on Medical Imaging* 39.8 (2020), pp. 2653–2663.

[109] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel. "Using trusted data to train deep networks on labels corrupted by severe noise." In: *Advances in neural information processing systems* 31 (2018).

[110] R. Yi, Y. Huang, Q. Guan, M. Pu, and R. Zhang. "Learning from pixel-level label noise: A new perspective for semi-supervised semantic segmentation." In: *IEEE Transactions on Image Processing* 31 (2021), pp. 623–635.

[111] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao. "Learning from weak and noisy labels for semantic segmentation." In: *IEEE transactions on pattern analysis and machine intelligence* 39.3 (2016), pp. 486–500.

[112] T. Liu and D. Tao. "Classification with noisy labels by importance reweighting." In: *IEEE Transactions on pattern analysis and machine intelligence* 38.3 (2015), pp. 447–461.

[113] H.-S. Chang, E. Learned-Miller, and A. McCallum. "Active bias: Training more accurate neural networks by emphasizing high variance samples." In: *Advances in Neural Information Processing Systems* 30 (2017).

[114] H. Zhang, X. Xing, and L. Liu. "Dualgraph: A graph-based method for reasoning about label noise." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9654–9663.

[115] M. Ren, W. Zeng, B. Yang, and R. Urtasun. "Learning to reweight examples for robust deep learning." In: *International conference on machine learning*. PMLR. 2018, pp. 4334–4343.

[116] S. Ye, D. Chen, S. Han, and J. Liao. "Robust Point Cloud Segmentation With Noisy Annotations." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.6 (2022), pp. 7696–7710.

[117] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. "On the shape of a set of points in the plane." In: *IEEE Transactions on information theory* 29.4 (1983), pp. 551–559.

[118] N. Abraham and N. M. Khan. "A novel focal tversky loss function with improved attention u-net for lesion segmentation." In: *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*. IEEE. 2019, pp. 683–687.

[119] A. Shrivastava, A. Gupta, and R. Girshick. "Training region-based object detectors with online hard example mining." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 761–769.

[120] M. Berman, A. R. Triki, and M. B. Blaschko. "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4413–4421.

[121] 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).

[122] W. A. Falcon. "Pytorch lightning." In: *GitHub* 3 (2019).

[123] 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.

[124] 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.

[125] 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.

[126] 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.