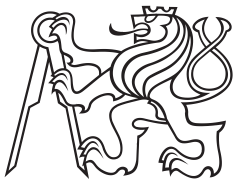
Bachelor Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Active Learning for Semantic Segmentation of Point Clouds

Aleš Kučera

Supervisor: MSc. Ruslan Agishev
Study Program: Cybernetics and Robotics
February 2023

Acknowledgements

I am writing to express my sincere gratitude to all those who have contributed to completing this thesis.

First and foremost, I extend my heartfelt thanks to my supervisor, MSc. Ruslan Agishev, for their invaluable guidance, encouragement, and support throughout my research. Their expertise and experience have greatly influenced my academic and professional growth.

I would also like to extend my sincere thanks to the employees of the faculty, who have provided me with the necessary resources and opportunities to pursue my academic goals.

I extend my gratitude to the Research Center for Informatics (RCI) for providing the resources to complete this thesis.

I am grateful to my classmates, friends, and colleagues who have offered their support and encouragement in various ways during my time as a student.

Finally, I would like to express my appreciation to my family for their love, support, and motivation throughout my academic journey.

I am deeply thankful to all of you and would like to acknowledge your invaluable contributions to completing this thesis.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 26, 2023

_____

Signature

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 26. května 2023

_____

podpis autora práce

Abstract

The primary objective of this thesis is to address the challenge of minimizing the annotation cost associated with point cloud datasets used in semantic segmentation tasks through active learning techniques.

Within this research, we present a novel active learning framework tailored explicitly for multi-view LiDAR-based datasets, which provide diverse viewpoints of objects. We thoroughly investigate the impact of incorporating multiple viewpoints on the performance of commonly employed uncertainty selection strategies in active learning.

Moreover, we introduce an innovative uncertainty selection strategy based on the variance of the model's outputs within our framework, offering a comparative analysis of its effectiveness against state-of-the-art methods. Additionally, we explore the significance of filtering out unreliable predictions when selecting annotated data for active learning.

To evaluate the efficacy of our proposed active learning framework, we conduct comprehensive experiments on widely used automotive datasets for LiDAR-based semantic segmentation. Through these evaluations, we effectively demonstrate how active learning can significantly improve the annotation efficiency of point cloud datasets.

Keywords: active learning, semantic segmentation, point clouds, machine learning

Supervisor: MSc. Ruslan Agishev

Abstrakt

Tato práce se zaměřuje na snížení nákladů spojených s anotací datasetů mračen bodů pro sémantickou segmentaci pomocí aktivního učení.

V rámci této studie představujeme nový přístup aktivního učení, který je speciálně navržen pro datasety se skeny ze senzoru LiDAR z více různých pohledů na stejný objekt. Důkladně zkoumáme, jak využití více pohledů ovlivňuje výkon běžně používaných strategií aktivního učení založených na nejistotě modelu. Kromě toho představujeme novou strategii výběru vzorků v rámci našeho přístupu a porovnáváme její účinnost s existujícími metodami. Dále se také zabýváme důležitostí filtrování nespolehlivých predikcí při výběru anotovaných dat.

Náš nový přístup aktivního učení je testován na široce používaných datasetech pro sémantickou segmentaci mračen bodů v automobilových aplikacích. Na základě výsledků evaluace úspěšně demonstrujeme, jak aktivní učení výrazně zlepšuje efektivitu anotace těchto datasetů pro sémantickou segmentaci.

Klíčová slova: aktivní učení, sémantická segmentace, mračna bodů, strojové učení

Překlad názvu: Aktivní učení pro sémantickou segmentaci mračen bodů

Contents

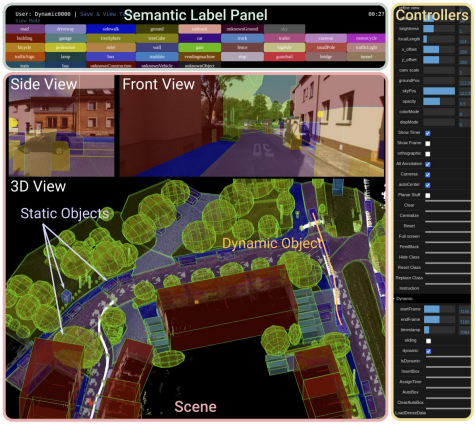
Chapter 1

Introduction

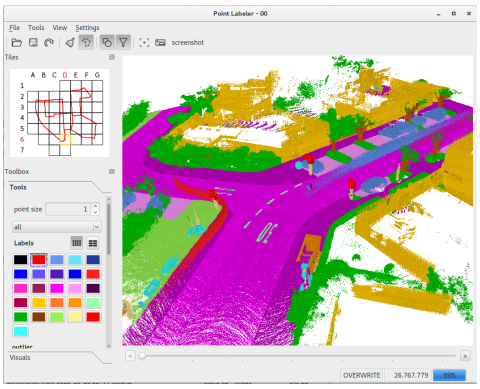
1.1 Motivation

Supervised learning plays a crucial role in machine learning by enabling accurate prediction of outcomes on new data. However, labeling data necessary for supervised learning can be time-consuming and expensive, mainly when dealing with large datasets. For example, labeling the ImageNet dataset [1], the most extensive image dataset available, has been estimated to require approximately 19 years [2] of labeling effort by a single person.

This labeling cost poses a significant obstacle, particularly in the context of large-scale LiDAR datasets used in applications such as autonomous driving and mobile robotics. A study [3] provides a detailed analysis of the time complexity involved in their labeling process. Although they managed to accelerate the typical labeling process by a factor of ten, they still had to invest approximately 1200 hours in labeling point clouds.



(a) : The KITTI-360 tool utilized for point cloud annotation [3].



(b) : The SemanticKITTI tool used for point cloud annotation [4].

Figure 1.1: Illustration of tools employed for point cloud annotation. Labeling each point for semantic segmentation tasks proves to be more challenging than labeling images.

In our own experience, we encountered the challenges of data labeling while

working on a project focused on estimating the traversability of a terrain for a mobile robot [1]. We initially attempted to label images with traversable, untraversable, and background labels in forest and town environments. However, we soon realized that the projection from the image to LiDAR data was not accurate enough for our purposes due to lidar-camera calibration inaccuracies and point occlusions observed in the camera field of view that were used to project the semantic labels. Consequently, we also had to label the LiDAR data, ensuring the labels were present in the point cloud format. This was essential because the path planning and control algorithms relied on these labels to plan the robot's movement. This process gave us firsthand insight into the intricacies and importance of accurate labeling in LiDAR-based applications. The traversability estimation dataset that contains labeled images and point clouds captured in the forest and suburban environment is available for download [2].

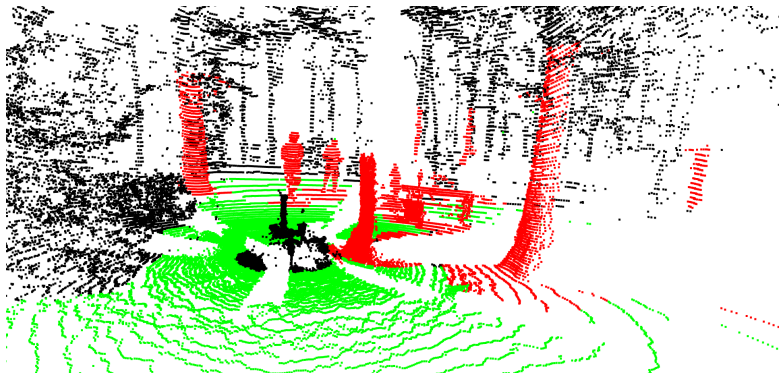


Figure 1.2: Labeled LiDAR scan depicting different categories within the scene. The red points (●) represent untraversable areas, including objects like trees or people and their boundaries. The green points (●) indicate traversable areas, indicating safe paths for navigation. The black points (●) are left unlabeled, as they represent semi-traversable regions or areas that are too distant and sparse, making it challenging to determine their traversability accurately.

Furthermore, we discovered that the challenges encountered in point cloud annotation extended beyond our project. Researchers in a separate study [4] reported that labeling a 100x100 meter tile required labelers between 1.5 to 4.5 hours, depending on the complexity of the labeling task. They recognized the difficulties inherent in annotating point clouds, highlighting how even experienced image labelers faced obstacles when navigating and transferring their knowledge to point cloud annotation. Additionally, they found that up to 15% of the labels had to be redone, further underscoring the complexities involved in accurately and efficiently labeling LiDAR data.

To address these challenges, active learning, a machine learning paradigm that aims to reduce the labeling effort, offers a promising solution. By iteratively selecting the most informative samples for annotation, active

[1] `https://github.com/ctu-vras/traversability_estimation`

[2] `http://subtdata.felk.cvut.cz/robingas/data/traversability_estimation/TraversabilityDataset/supervised/`

learning significantly reduces the overall annotation cost compared to a fully supervised approach.

1.2 Contributions

In this thesis, our aim is to make the following contributions to the field of active learning for LiDAR-based datasets:

1. We propose a novel active learning pipeline designed explicitly for sequence LiDAR-based datasets containing multiple viewpoints of the same object. This pipeline improves the reliability of the uncertainty score in the uncertainty selection strategies and aligns with the state-of-the-art annotation workflow [3].

2. We investigate the impact of filtering unreliable points that can contribute to the model's uncertainty, such as distant and sparse points. This additional filtering step aims to improve the quality of the training data and enhance the performance of the active learning pipeline.

3. To further enhance our active learning pipeline, we propose an additional active learning selection strategy called Viewpoint Variance, inspired by Viewpoint Entropy [5]. This strategy aims to incorporate the variability of viewpoints within the selection process, enabling the model to better generalize to different perspectives.

By exploring these aspects, our research aims to improve the effectiveness and efficiency of active learning for point cloud semantic segmentation. We make the source codes of the active learning pipeline [3] publicly available as a part of the Bachelor's Thesis.

1.3 Thesis Outline

The thesis is structured as follows. Chapter 2 provides a general insight into active learning and standard uncertainty methods. Following that, Chapter 3 discusses the existing research and approaches in active learning for point cloud semantic segmentation. In Chapter 4, we introduce our proposed active learning pipeline, which includes the contributions mentioned in the previous section. Chapter 5 discusses the experimental settings chosen for the experiments, and Chapter 6 presents the results. Finally, Chapter 7 summarizes the work done in this thesis and proposes future directions for further improvement.

[3] https://github.com/aleskucera/MuVAL

Chapter 2

Theoretical Background

This chapter is dedicated to providing a comprehensive understanding of active learning and the strategies employed within the scope of this thesis. The first section serves as a notational explanation that will be referred to later in the text. In the second section, we establish a parallel between active learning and human learning, as discussed in work [6]. Furthermore, we present a practical machine learning example to illustrate the core concept. Subsequently, an exploration of fundamental uncertainty selection strategies, namely Confidence (**CONF**), Margin (**MAR**), and Entropy (**ENT**), is undertaken. Additionally, we delve into utilizing the model's epistemic uncertainty (**EPI**) approximation as a selection strategy.

2.1 Notation

To establish the notation for the function representing the neural network's forward pass, we define it as follows:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \theta, \mathbf{w}) = \mathbf{f}_\theta(\mathbf{x}, \mathbf{w}) \tag{2.1}$$

Here, $\mathbf{x}$ denotes the input vector, θ represents the parameters that define the model architecture within a broad class of functions, and $\mathbf{w}$ captures the mappings from these functions to the desired input. It should be noted that we will omit the weights $\mathbf{w}$ in cases where their presence is unnecessary.

In various instances, it becomes necessary to represent the output as a probability distribution. To specifically emphasize a scenario that describes a model output as a probability distribution given input $\mathbf{x}$ and weights $\mathbf{w}$, we utilize the following notation:

$$P_\theta(\mathbf{y}|\mathbf{x}, \mathbf{w}). \tag{2.2}$$

2.2 Active Learning in Education

To enhance the comprehension of active learning in the context of machine learning, we draw a parallel to the educational system, where a similar distinction can be made between two types of learning techniques:

- **Passive Learning**: This type of learning is characterized by low student engagement, where the student is just an observer. Examples of passive learning include lecture-style teaching. It is important to note that passive learning can be more beneficial than active learning when the student has little or no prior knowledge.
- **Active Learning**: Unlike passive learning, active learning requires students to engage in activities that enable them to acquire new knowledge. Examples of active learning techniques include project-based learning or classroom discussions. The idea behind active learning is that students build upon their existing knowledge to acquire new knowledge. However, active learning requires the teacher to understand the student's current knowledge level better and necessitates tailored activities that suit their learning needs. Despite this challenge, active learning can lead to a more profound understanding of the subject matter by the student.

2.3 Active Learning in Machine Learning

Similar to the preceding section on active learning in education, we can draw a parallel in machine learning, differentiating between passive and active learning. In this context, the model serves as the student, while the dataset assumes the role of the learning material.

- **Passive Learning**: In this traditional machine learning model, the dataset is prepared beforehand and doesn't change during the training process based on the model's feedback. This is similar to passive learning in education, where the student observes a lecture without asking questions.
- **Active Learning**: In contrast to passive learning, active learning involves the algorithm selecting the data it wants to learn from and constructing the dataset step-by-step. This selection process is guided by the goal of acquiring new knowledge or reducing uncertainty based on the model's experience learned from previous data samples. This process is more time-consuming than passive learning because the network has to be fine-tuned after each step, and must acquire a better understanding of the data to choose the most valuable samples for labeling. However, this approach can save the annotator's time by providing fewer samples to label. The learner can select data from a pool or sequence to be labeled by an annotator or create instances that require labeling. Active learning in machine learning can be understood in the philosophy of constructivism, similar to active learning in education.

As mentioned, the fundamental hypothesis of active learning is that if the learning algorithm can choose the data from which it learns, it will perform better with less training. This is because active learning systems attempt to overcome the labeling bottleneck by asking queries in the form of unlabeled

instances to be labeled by an oracle, such as a human annotator. In this way, the active learner aims to achieve high performance using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data.

Active learning is well-motivated in many modern machine learning problems where data may be abundant, but labels are scarce or expensive. This is particularly relevant in natural language processing and computer vision, where labeling large datasets is often laborious and time-consuming.

2.4 Active Learning Scenarios

According to the study [7], there are three common active learning scenarios: membership query synthesis, stream-based selective sampling, and pool-based sampling. In this section, we will focus on the pool-based scenario.

2.4.1 Pool-based Active Learning Framework

In the pool-based active learning scenario, a substantial pool of unlabeled data $\mathcal{U}$ is available, from which samples are selected for labeling and subsequently added to the labeled data pool $\mathcal{L}$. The machine learning model is initially

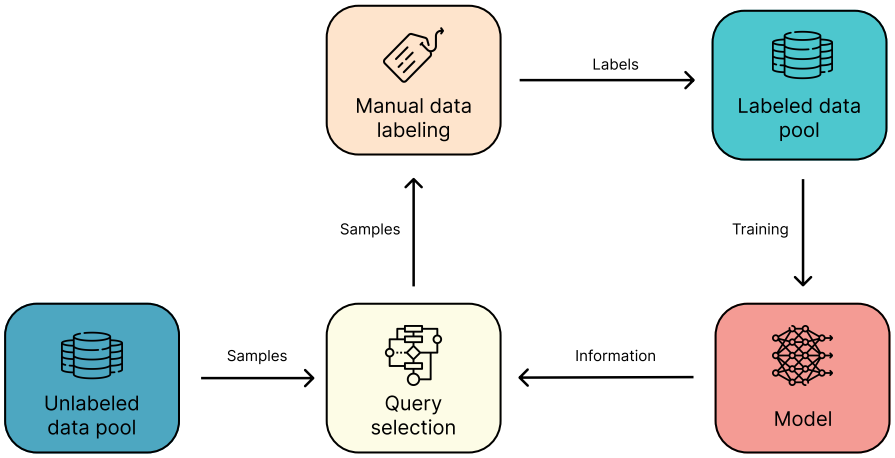


Figure 2.1: Typical pool-based active learning pipeline. At the beginning of the process, there is a large pool of unlabeled data. In each iteration, a portion of the data is selected for labeling based on the predictions of the machine learning model. The labeled data is then used to improve the model's accuracy, and the process repeats until the desired level of accuracy is achieved.

trained on a small set of labeled examples in the pool-based active learning framework. Subsequently, according to a desired metric increase, the model selects the most informative samples from the remaining unlabeled pool to be labeled by an oracle. This selection process may involve choosing samples for which the model exhibits the least certainty in its predictions, among

other criteria. The newly labeled data is then integrated into the training set, and the model is retrained using the updated set of labeled examples. This iterative process is repeated, leading to an increasingly accurate model with each labeling round.

Many different active learning selection strategies can be employed to select the most informative examples for labeling. These include uncertainty sampling, query-by-committee, and density-based sampling, among others [7]. The choice of strategy depends on the dataset's characteristics and the specific problem being addressed.

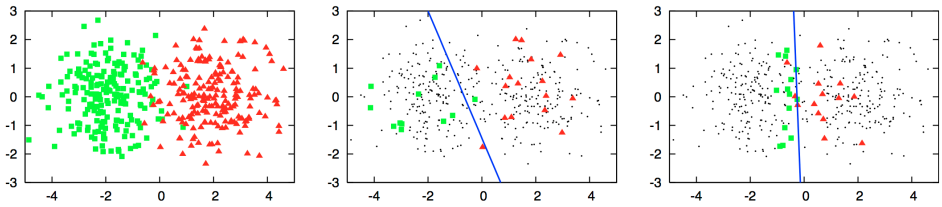


Figure 2.2: An illustrative example of pool-based active learning [7]. We have a dataset of 400 instances evenly sampled from two class Gaussians, and we aim to train a logistic regression model for a classification task using only 30 labeled instances. The left-hand image shows the ground truth for all instances. The middle picture represents a random selection of 30 instances, achieving an accuracy of 70%. However, the last image demonstrates the significant improvement that can be made by actively selecting examples using uncertainty sampling, which performs an accuracy of 90%.

2.5 Query Selection Strategies

In all active learning scenarios, the evaluation of the informativeness of unlabeled instances plays a crucial role. To facilitate this process, query strategies are employed, which significantly impact the effectiveness of active learning techniques.

Active learning strategies can be broadly categorized into two types [8]. The first type focuses on *diversity criteria*, aiming to provide the model with a diverse range of data samples. The second type employs *uncertainty strategies*, which involve presenting the model with data samples exhibiting the highest level of uncertainty. In this thesis, we concentrate on **uncertainty strategies**.

Subsequently, we aim to specify the most informative sample based on a particular criterion. To facilitate this, we employ the notation x^*_A, where A represents the criterion upon which we define the informativeness. It is important to note that although the sample x is generally a vector, the provided formulas for selection are simplified to the scalar x. Generalization is achieved by computing the criterion score for each element separately and subsequently calculating the mean of these scores.

2.5.1 Common Uncertainty Strategies

One of the widely adopted query frameworks is uncertainty sampling. In uncertainty sampling, an active learner selects instances with the least confidence in assigning labels.

In the case of binary classification, samples are chosen based on the proximity of the model's prediction to 0.5. For multiclass classification, samples can be selected using the *confidence* (**CON**) of prediction. The selection is determined by the following formula

$$x^*_{\text{CON}} = \underset{x}{\operatorname{argmin}} \left(p_\theta(\hat{y}_i|x_i)\right) \tag{2.3}$$

which can be interpreted as the model's belief that it will mislabel an instance.

To consider more about the whole model's output distribution, we can select based on the difference between the first two most probable classes. This method is called *margin sampling* (**MAR**) and can be written as

$$x^*_{\text{MAR}} = \underset{x}{\operatorname{argmin}} \left(p_\theta(\hat{y}_1|x) - p_\theta(\hat{y}_2|x)\right), \tag{2.4}$$

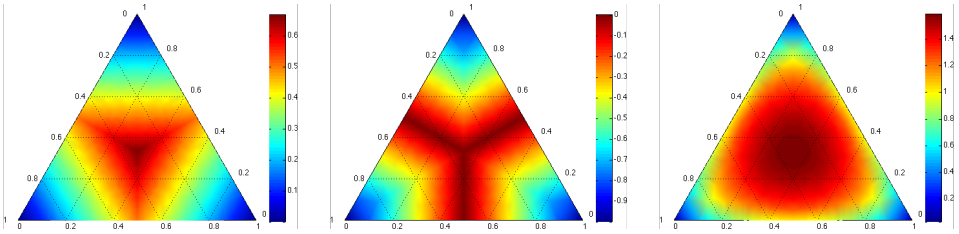
where $\hat{y}_1$ and $\hat{y}_2$ are the first and the second probable class respectively. This method aims to steer apart close predictions and learn the boundary between individual classes.

A more general approach to calculating uncertainty is to use *entropy* (**ENT**), which measures the level of uncertainty in the probability distribution. This method can be applied to any label set size and is expressed mathematically

$$x^*_{\text{ENT}} = \underset{x}{\operatorname{argmax}} \left(-\sum_{i=1}^{c} p_\theta(y_i|x) \log p_\theta(y_i|x) \right) \tag{2.5}$$

The relationship between the different uncertainty measures (2.3), (2.4) and (2.5) can be visualized in Figure 2.3. The most informative instance can be found at the center of the triangle in all cases, as this is where the posterior label distribution is the most uniform, indicating the highest level of uncertainty under the model. On the other hand, the least informative instances are located at the three corners of the triangle, where one of the classes has an extremely high probability and little model uncertainty.

Regarding selecting an appropriate uncertainty measure, entropy is generally suitable when the objective function is to minimize log-loss. In contrast, the other two measures, particularly margin sampling, may be more appropriate when the goal is to reduce classification error. They prioritize instances to help the model discriminate among specific classes [7].



(a) : Confidence (**CON**) **(b) :** Margin (**MAR**) **(c) :** Entropy (**ENT**)

Figure 2.3: Heatmaps illustrating the query behavior of common uncertainty measures in a three-label classification problem. [7]

2.5.2 Epistemic Uncertainty as Selection Strategy

Bayesian Neural Networks (BNNs) learn the approximate distribution of weights to generate uncertainty estimates, which reflect prediction confidences. Within this framework, two types of uncertainties are commonly distinguished [9]: aleatoric uncertainty, which quantifies the intrinsic uncertainty stemming from the observed data, and epistemic uncertainty, which arises from the model's uncertainty. Epistemic uncertainty is typically estimated by inferring the posterior weight distribution through Monte Carlo sampling. Unlike aleatoric uncertainty, which captures irreducible noise in the data, epistemic uncertainty can be reduced by gathering more training data. For instance, segmenting an object with a relatively small number of training samples may result in high epistemic uncertainty, while high aleatoric uncertainty may occur at segment boundaries or for distant and occluded objects due to inherent sensor noise. Bayesian modeling enables the estimation of both types of uncertainty.

To calculate the epistemic uncertainty, we can employ the Monte Carlo Dropout technique described in the study [9]. By performing n forward passes with independently sampled dropout masks, which create a set of different model structures

$$\Theta = \{\theta_1, ..., \theta_n\}, \tag{2.6}$$

we obtain a set of model outputs for each input x:

$$\mathcal{F}_{\text{MC}} = \{\mathbf{f}(x, \theta_1), ..., \mathbf{f}(x, \theta_n)\} \tag{2.7}$$

By applying the formula described in the referenced study [9], we can compute the *dropout variance vector* as a metric of epistemic uncertainty. This variance vector, denoted as $\boldsymbol{\nu}$, is derived from the input x and multiple model structures modified by dropout Θ using formula

$$\boldsymbol{\nu}(x, \Theta) = \text{Var}(\mathcal{F}_{\text{MC}}) = \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{f}(x, \theta_i) - \frac{1}{n} \sum_{j=1}^{n} \mathbf{f}(x, \theta_j) \right)^2. \tag{2.8}$$

The variance vector has a size of c, where c represents the number of classes. We compute the mean of the values in this vector to obtain an overall

epistemic uncertainty score (**EPI**). Consequently, the most informative sample based on this is the one with a maximum value:

$$x^*_{\text{EPI}} = \underset{x}{\text{argmax}} \left(\frac{1}{c} \sum_{i=1}^{c} \nu_i(x, \Theta) \right). \tag{2.9}$$

In this equation, ν_i refers to the i-th element of the variance vector associated with the input x and can be interpreted as a variance of the predictions for a certain class.

Chapter 3

Related Work

Numerous approaches have been proposed to mitigate the annotation cost, and these methods can also be adapted for point clouds semantic segmentation datasets. One approach involves self-supervision [10, 11, 12], where deep learning models are pre-trained and fine-tuned using limited annotations. Another option is domain transfer [13, 14], which leverages the availability of large existing datasets. Additionally, weak supervision techniques have been employed [15, 16, 17], where only a subset of points, regions, or scenes within the point cloud are annotated, providing partial or noisy labels. Finally, active learning techniques [18, 19, 20, 8] aim to identify the most informative points, regions, or scenes within the data for annotation, improving weakly supervised learning by selecting samples that would maximize the learning gain.

The current state-of-the-art method for active learning on LiDAR-based datasets is ReDAL [18]. This approach combines uncertainty and diversity by selecting regions based on factors such as the entropy of model predictions, color discontinuity, and surface variation. Furthermore, it incorporates diversity-aware selection to ensure the inclusion of the most diverse regions.

It was observed that traditional uncertainty methods (2.3), (2.4) and (2.5) did not achieve satisfactory performance when applied to LiDAR datasets [18, 8]. However, it is worth noting that these methods have shown success in the context of 2D images [5]. This performance discrepancy could be attributed to the sparsity of LiDAR points in certain regions. In such cases, where the points are sparse or regions are labeled as void, the model's uncertainty estimation may not be reliable or accurate. This suggests that LiDAR data's sparsity could contribute to the limited effectiveness of uncertainty-based approaches in the semantic segmentation of LiDAR data.

In reference to Section 1.2, our aim is to investigate potential solutions to the poor performance of traditional uncertainty methods in point cloud semantic segmentation. Through our research, we aim to contribute new insights and advancements in this field.

Chapter 4

Method

4.1 Overview

This section overviews our customized active learning pipeline, designed for LiDAR-based semantic segmentation. The pipeline aims to improve the efficiency and reliability of active learning by incorporating uncertainty estimation and considerations for sequence datasets. Figure 4.1 presents a visual representation of the entire process.

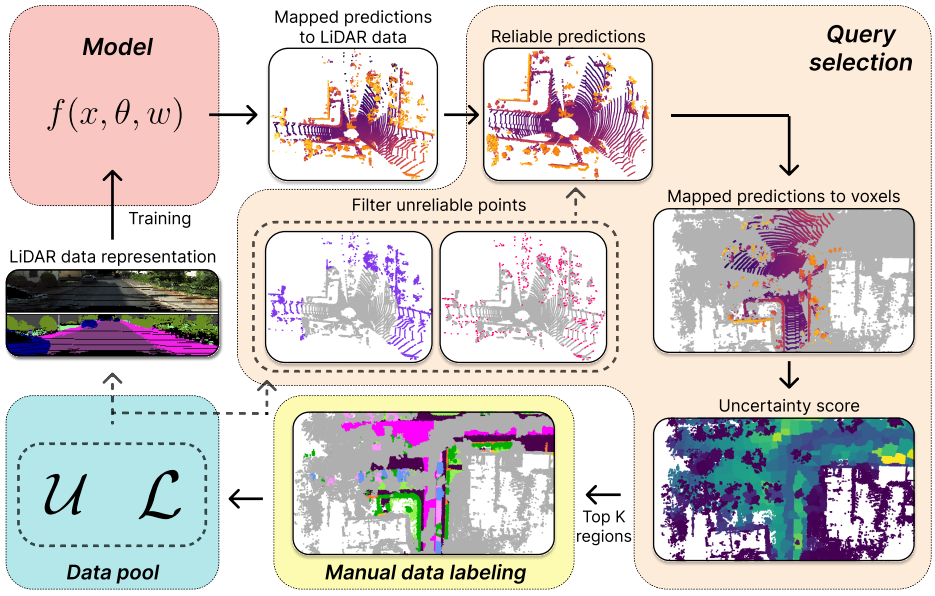


Figure 4.1: Illustration of proposed active learning pipeline. The pipeline involves training a model with a labeled dataset, filtering reliable points with predictions, fusing and voxelizing the reliable points, calculating uncertainty scores for regions, and selecting regions with high uncertainty for labeling.

The pipeline begins with training the model using a labeled dataset $\mathcal{L}$. Once trained, the model generates predictions for each scan in the dataset. To ensure the reliability of the points, a filtering process is applied to retain only those with reliable predictions. These reliable points are then fused

and voxelized, forming regions that represent clusters of neighboring voxels. An uncertainty score is calculated for each region, providing an indication of the level of uncertainty associated with it. The uncertainty scores are used to select regions with the highest uncertainty for labeling, facilitating an iterative active learning process.

4.1.1 Comparison: Our Pipeline vs. ReDAL

In this subsection, we compare our proposed pipeline with the ReDAL (Region-based and Diversity-Aware Active Learning) pipeline 4.2 introduced in [18]. The ReDAL pipeline aims to address the challenges of uncertainty estimation in LiDAR scans by incorporating color discontinuity and structural complexity into the selection score. However, our approach is tailored explicitly for uncertainty methods and sequence datasets, such as KITTI-360 [21] and SemanticKITTI [22], and offers several distinct advantages.

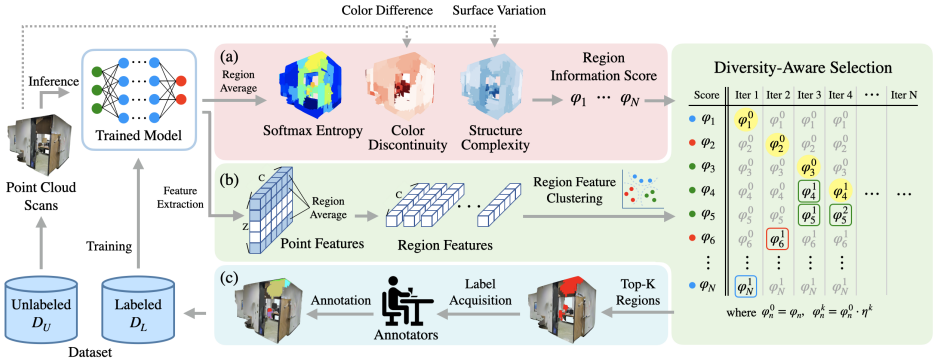
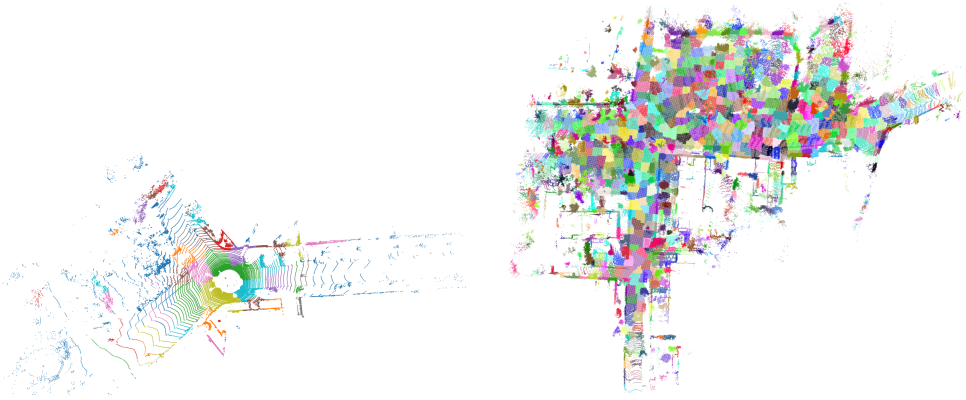


Figure 4.2: ReDAL (Region-based and Diversity-Aware Active Learning) pipeline. [18]

Unlike ReDAL, which operates on individual scans, our pipeline leverages fused point clouds generated from multiple scans. This allows us to benefit from multiple model predictions obtained from different viewpoints, resulting in more reliable computation of model uncertainty for each point in space. Moreover, labeling in the fused point cloud aligns better with state-of-the-art labeling methods [21], as it simplifies the annotation process by providing denser point coverage and eliminates redundancy in labeling the same position in space for each scan individually.

By utilizing the unique features of sequence datasets, we aim to obtain more reliable uncertainty scores. Additionally, we explore the effectiveness of filtering out unreliable points before calculating the uncertainty score, further improving the accuracy of our pipeline.

To illustrate the difference between selecting regions in an individual scan and the fused point cloud, refer to Figure 4.3. The sparsity of individual scans makes it challenging to approximate objects accurately while selecting regions in the fused point cloud propagates label information to all scans containing points within those regions.



(a) : Partitioning of the LiDAR scan into regions [18].

(b) : Partitioning of the fused point cloud from multiple scans.

Figure 4.3: Visualization highlights the difference between partitioning and selecting regions in an individual scan and the fused point cloud. It is evident that due to the sparsity of the scan, it is challenging to approximate objects in the scan. Additionally, selecting a region in the fused cloud propagates label information to all scans containing points within that region.

For a detailed explanation of the partitioning process, please refer to Section 5.2.2.

4.2 Point Cloud Filters

In this section, we present two filters that are integrated into our active learning pipeline to enhance the reliability of the active learning selection strategies. The first filter (**DIST**) is designed to remove distant points captured by the LiDAR sensor. The rationale behind this filter is that as the distance increases, sparsity becomes more prominent, and sensor noise becomes more noticeable. Let $\mathcal{P}$ represent the set of points in the scan. The reliable points $\mathcal{R}$, determined by applying the **DIST** filter, therefore $\mathcal{R}_\rho^{\text{DIST}}$ can be defined as follows:

$$\mathcal{R}_\rho^{\text{DIST}} = \{\mathbf{p} \mid \mathbf{p} \in \mathcal{P},\ \|\mathbf{p}\| < \rho\}, \tag{4.1}$$

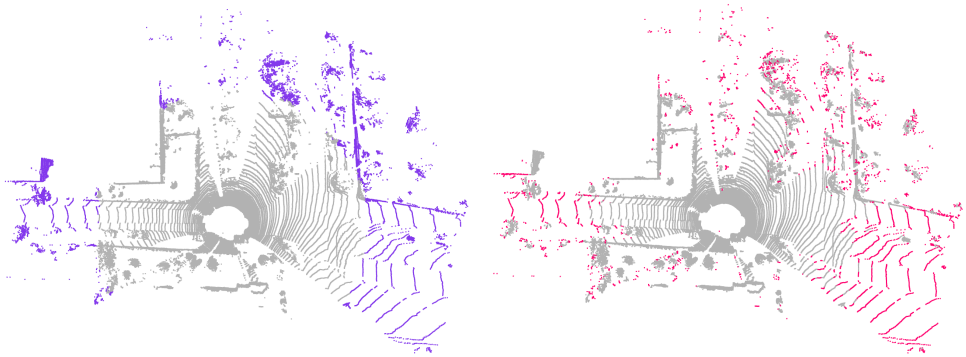
where ρ is the radius within which points are considered reliable.

The second filter (**NN**) aims to address the sparsity issue by removing points that do not have a sufficient number of nearest neighbors within a specified radius. Let $\mathcal{N}_\varepsilon^{\mathbf{p}}$ denote the set of points from the same scan that are within a radius ε of point $\mathbf{p}$:

$$\mathcal{N}_\varepsilon^{\mathbf{p}} = \{\mathbf{n} \mid \mathbf{n}, \mathbf{p} \in \mathcal{P},\ \|\mathbf{p} - \mathbf{n}\| < \varepsilon,\ \mathbf{n} \neq \mathbf{p}\}. \tag{4.2}$$

We can define the reliable points $\mathcal{R}_{\varepsilon,k}^{\text{NN}}$, which have at least k nearest neighbors within the radius ε, as:

$$\mathcal{R}_{\varepsilon,k}^{\text{NN}} = \{\mathbf{p} \mid \mathbf{p} \in \mathcal{P},\ |\mathcal{N}_\varepsilon^{\mathbf{p}}| \geq k\}. \tag{4.3}$$



(a) : Filtering reliable points based on distance from the LiDAR sensor. ($\rho = 30$)

(b) : Filtering reliable points based on the number of nearest neighbors within a radius. ($\varepsilon = 0.1,\ k = 20$)

Figure 4.4: Visualization highlighting the difference between the proposed filters. The purple points (●) are marked as unreliable by the **DIST** filter (4.1), and the pink points (●) are marked as unreliable by **NN** 4.3 filter.

4.3 Uncertainty Score: Viewpoint Variance

In this section, we present our proposed uncertainty score called *Viewpoint Variance* (**VV**). This score enhances the active learning process by considering the variance in predictions across different viewpoints of the same object in LiDAR datasets. By leveraging multiple viewpoints, we gain a better understanding of the uncertainty associated with each object.

To calculate the Viewpoint Variance, we utilize multiple viewpoints by providing n different viewpoints of the same object, resulting in a set of model inputs $\mathcal{X} = \{x_1, ..., x_n\}$. This, in turn, generates a set of model outputs $\mathcal{F}_{\mathrm{V}} = \{\mathbf{f}(x_1, \theta), ..., \mathbf{f}(x_n, \theta)\}$.

Using these model outputs, we calculate the *viewpoint variance vector* $\boldsymbol{\vartheta}$ based on the model architecture θ and multiple inputs $\mathcal{X}$. The viewpoint variance vector is defined as the variance of the model outputs:

$$\boldsymbol{\vartheta}(\mathcal{X}, \theta) = \mathrm{Var}(\mathcal{F}_{\mathrm{V}}) = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{f}(x_i, \theta) - \frac{1}{n}\sum_{j=1}^{n}\mathbf{f}(x_j, \theta)\right)^2. \tag{4.4}$$

Here, ϑ_i represents the input variance of the i-th class, calculated using the viewpoints $\mathcal{X}$ and model θ. The variance vector has a size of c, where c corresponds to the number of classes.

To select the most informative sample based on the viewpoint variance vector, we compute the viewpoint variance score (**VV**) as the mean of the individual elements in the variance vector. We determine the input sample with the highest overall viewpoint variance, given by:

$$\mathcal{X}^*_{\mathrm{VV}} = \underset{\mathcal{X}}{\mathrm{argmax}}\left(\frac{1}{c}\sum_{i=1}^{c}\vartheta_i(\mathcal{X}, \theta)\right). \tag{4.5}$$

In our approach, we consider these multiple inputs as a sample, assuming that they are different viewpoints of the same object with the same label. We define $\mathcal{X}$ as a voxel in our pipeline and assume that each point within the voxel corresponds to a different LiDAR scan viewpoint.

By maximizing the mean of the variances of the predictions for points within the voxel, we select the most informative voxels in the fused cloud.

Chapter 5

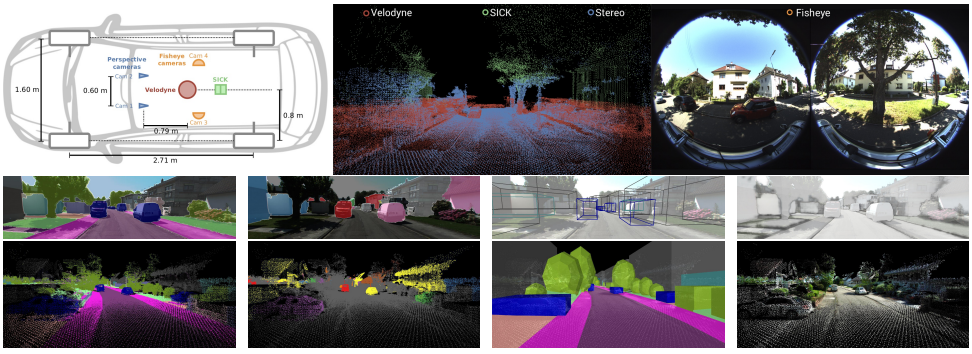
Experimental Settings

In this section, we present the experimental settings employed in our study to evaluate the performance and effectiveness of our proposed methods. We provide an overview of the datasets and their preprocessing, the model architecture selected, the loss function employed, and the augmentations applied. These experimental settings were carefully chosen to ensure robust and accurate results while addressing the specific challenges posed by LiDAR data analysis. By describing the critical components of our experimental setup, we lay the foundation for the subsequent evaluation and analysis of our proposed approaches.

5.1 Datasets

We have carefully selected two widely used datasets, namely KITTI-360 [21] and SemanticKITTI [22], for our experimental evaluations. These datasets offer rich and diverse LiDAR data capturing various real-world scenarios, enabling us to comprehensively assess our proposed methods' performance and generalization capabilities.

1. **KITTI-360:** The KITTI-360 dataset is a collection of panoramic images and corresponding LIDAR scans captured by a moving vehicle in the suburbs of Karlsruhe, Germany. The dataset offers diverse environments,



(a) : Semantic **(b) :** Instance **(c) :** Bounding Box **(d) :** RGB

Figure 5.1: Overview of the KITTI-360 dataset.

including urban, residential, and rural areas, providing valuable training data for models operating in various scenarios. The dataset comprises multiple sensor modalities, such as a perspective stereo camera, a pair of fisheye cameras, a Velodyne, and a SICK laser scanning unit. These modalities enable the collection of data that allows for 360-degree scene perception. The dataset also includes comprehensive annotations, including consistent semantic and instance labels for every 2D image pixel and 3D point, making it an invaluable resource for developing and evaluating machine learning models.

2. **SemanticKITTI:** The SemanticKITTI dataset provides a valuable collection of LIDAR scans with comprehensive point-wise annotations, covering the full 360-degree field-of-view commonly utilized in automotive applications. This dataset encompasses a diverse range of urban and rural environments, offering semantic labels for objects such as cars, pedestrians, and buildings. Due to its size and diversity, the SemanticKITTI dataset has become widely adopted as a benchmark for assessing the performance of semantic segmentation models.

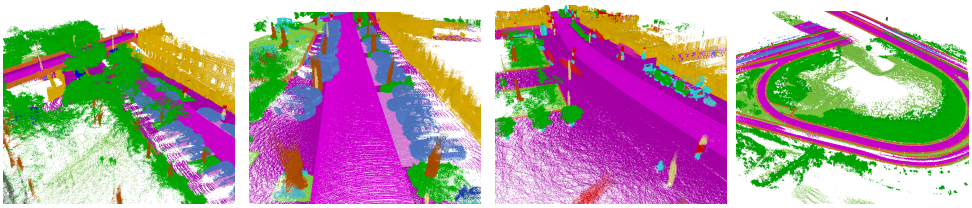


Figure 5.2: Overview of the SemanticKITTI dataset.

5.2 Dataset Adjustment and Partition

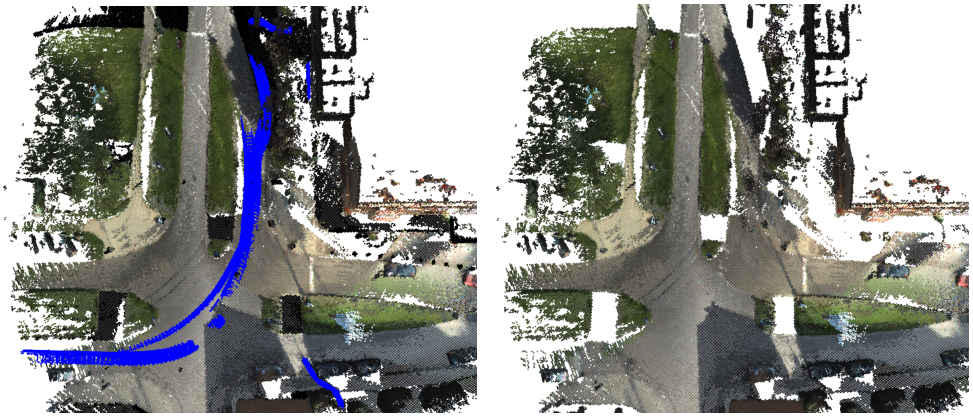
This thesis applied certain modifications to the KITTI-360 and SemanticKITTI datasets to tailor them for the active learning framework and enable efficient experimentation.

5.2.1 Data Preprocessing and Modification

In order to utilize multiple viewpoints for uncertainty estimation, dynamic object removal was performed. The dynamic objects introduce inconsistencies in the time dimension, making it impossible to calculate reliable uncertainty scores without accounting for the motion of these objects. Hence, dynamic points were excluded from the dataset to maintain the integrity of uncertainty estimation.

Void class removal involved handling points without any annotation provided. While these points can contribute to the geometric background of annotated points during training, they pose challenges when analyzing the regions the model selects. Without annotations, it becomes difficult to determine which regions the model had difficulty recognizing based solely on

statistical information. Therefore, void class points were removed to facilitate the identification of problematic regions selected by the model.



(a) : Original scene. **(b) :** Filtered scene.

Figure 5.3: Visualization of the dynamic points (●) and points labeled as void (●) and their removal.

To facilitate faster superpoint calculation and conserve memory resources, the sequence fused clouds were split into subsequences of approximately 200 scans. This partitioning strategy allowed for more efficient scoring calculations for each subsequence. Instead of storing predictions for each point in memory, the subsequences were treated as separate, non-continuous objects, enabling the calculation of scores on a per-subsequence basis.

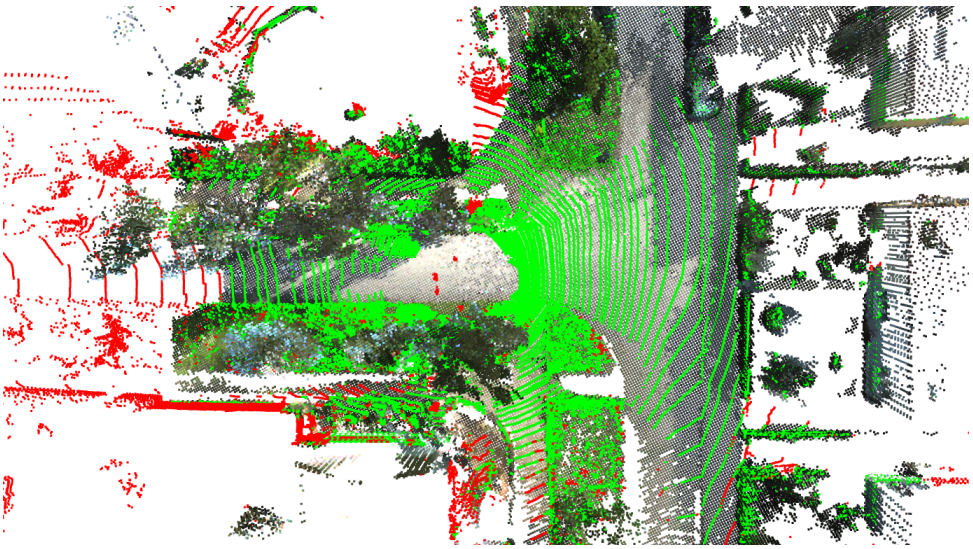


Figure 5.4: In the resulting scans, the removed points are represented by red markers (●), while the retained points are indicated by green markers (●). This selective removal of points helps refine the dataset and focus on the relevant information for subsequent analysis and model training.

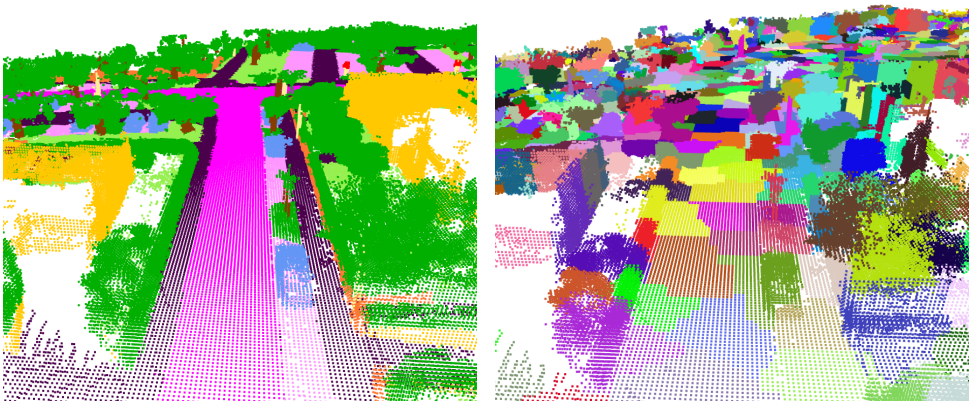
Due to the large size of the KITTI-360 and SemanticKITTI datasets, it

was necessary to reduce their overall length for practical reasons. During the experiments, the datasets were shortened to only include sequence 3 for KITTI-360 and sequences 3 and 4 for SemanticKITTI. This reduction in dataset size enabled faster experimentation, allowing meaningful results to be obtained within hours rather than days.

By applying these dataset preprocessing and partitioning techniques, the datasets were modified to suit the requirements of the active learning approach proposed in this thesis. These modifications aimed to enhance the efficiency of the framework and facilitate the analysis of selected regions in the point cloud data.

5.2.2 Partitioning the Fused Cloud into Regions

To facilitate the active learning approach proposed in this thesis, the fused clouds obtained from the preprocessing steps are partitioned into regions, commonly known as superpoints or supervoxels [23]. These regions serve as the fundamental units for subsequent analysis and selective sampling.



(a) : Original scene. **(b) :** Partitioned scene.

Figure 5.5: Visualization of the partitioned scene into regions.

Partitioning the fused clouds into regions is crucial for several reasons. Firstly, it allows us to handle the point cloud data at a higher level of abstraction, enabling more efficient processing and analysis than operating on individual points. By grouping together neighboring points that exhibit similar properties, such as proximity and geometric attributes, we can extract meaningful regions that represent distinct objects or surfaces in the scene.

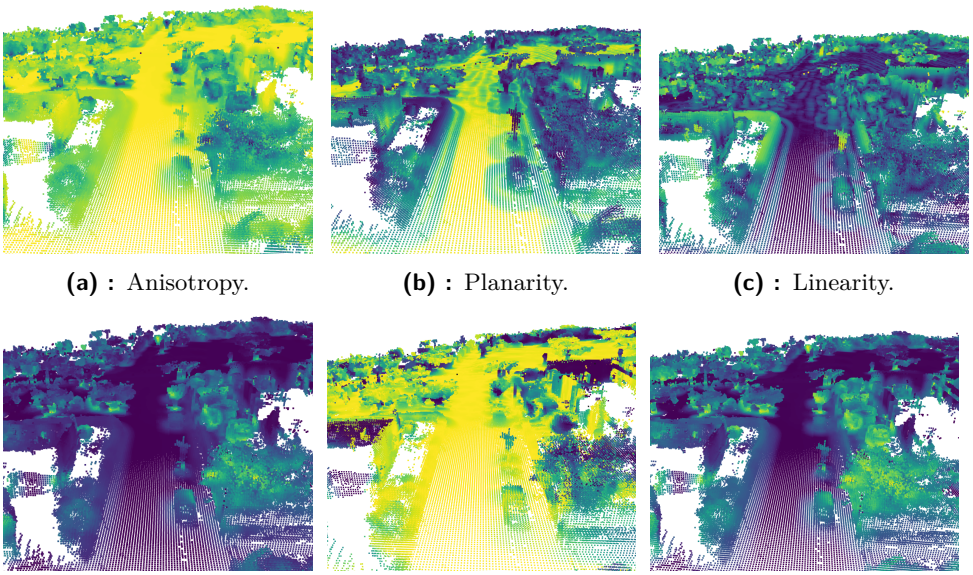
To partition the fused clouds, we employ a method that involves computing geometric features based on the 3D covariance matrix, commonly called the 3D structure tensor [24]. The 3D structure tensor provides valuable information about the local geometry of the point cloud, which can be extracted through eigenvectors and eigenvalues.

The 3D structure tensor, denoted as $\mathbf{X}$, captures the spatial distribution of points within a neighborhood and allows us to characterize their geometric properties. By calculating the eigenvectors and eigenvalues of $\mathbf{X}$, we can

derive geometric features that describe various aspects of the point cloud's structure.

Feature	**Expression**
Anisotropy	$(\lambda_1 - \lambda_3)/\lambda_1$
Planarity	$(\lambda_2 - \lambda_3)/\lambda_1$
Linearity	$(\lambda_1 - \lambda_2)/\lambda_1$
Sphericity	λ_3/λ_1
Verticality	$1 - \lvert(0,0,1) \cdot \mathbf{e_3}\rvert$
Surface Var.	$\lambda_3/(\lambda_1+\lambda_2+\lambda_3)$

Table 5.1: Geometric features calculated from the 3D structure tensor. These features capture various aspects of the point cloud's local geometry. The formulas describe the computation of each feature using the eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ and eigenvectors $(\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3})$ derived from the 3D covariance matrix $\mathbf{X}$.



(a) : Anisotropy. **(b) :** Planarity. **(c) :** Linearity.

(d) : Sphericity. **(e) :** Verticality. **(f) :** Surface Variation.

Figure 5.6: Visualization of the geometric features used for subsequence partitioning.

The computed geometric features include anisotropy, linearity, planarity, sphericity, verticality, and surface variation. These features provide insights into the local shape characteristics, such as the presence of elongated structures, planar surfaces, or spherical regions. Additionally, the position information of the points and, if available, the RGB values are incorporated to enrich the feature representation.

We use a graph-based algorithm [23] to partition the point cloud into

superpoints based on the computed geometric features. This algorithm leverages the connectivity between points and the similarity of their geometric properties to group them into coherent regions. By constructing a graph representation of the point cloud, where nodes correspond to points and edges capture their pairwise relationships, the algorithm identifies clusters of points with similar geometric features.

The resulting regions represent meaningful segments within the point cloud, encapsulating distinctive objects or surfaces. This graph-based approach provides a flexible and efficient way to partition the point cloud, enabling subsequent analysis and selective sampling of specific regions of interest.

5.3 Model

Deep learning models for 3D semantic segmentation using point clouds have become an important research area in recent years. LiDAR point clouds, which are direct reflections of real-world scenes, have unique characteristics that bring extra difficulties in learning, including diversity and disorder [25]. Therefore, a good representation is needed for efficient and effective LiDAR point cloud processing.

5.3.1 LiDAR Data Representation Selection

Various representations for LiDAR data have been proposed, including point view [26, 27], voxel view [28, 29], and multi-view fusion [30]. However, these methods often require computationally intensive neighborhood search, 3D convolution operations, or multi-branch networks, which can be inefficient during training and inference stages.

View	Formation	Complexity	Representative
Raw Points	Bag-of-Points	$\mathcal{O}(N \cdot d)$	RandLA-Net
Range View	Range Image	$\mathcal{O}(\frac{H \cdot W}{r^2} \cdot d)$	SqueezeSeg
Bird's Eye View	Polar Image	$\mathcal{O}(\frac{H^2 \cdot W}{r^2} \cdot d)$	PolarNet
Voxel (Dense)	Voxel Grid	$\mathcal{O}(\frac{H \cdot W \cdot L}{r^3} \cdot d)$	PVCNN
Voxel (Sparse)	Sparse Grid	$\mathcal{O}(N \cdot d)$	MinkowskiNet
Voxel (Cylinder)	Sparse Grid	$\mathcal{O}(N \cdot d)$	Cylinder3D
Multi-View	Multiple	$\mathcal{O}((N + \frac{H \cdot W}{r^2}) \cdot d)$	AMVNet

Table 5.2: Comparisons among different LIDAR representations [31]

Projection-based representations, such as the range view and bird's eye view, have been investigated for 3D semantic segmentation using point clouds, and various fusion approaches have been proposed. However, these representations have drawbacks that must be addressed. For example, the bird's eye

view introduces quantization error when dividing the space into voxels or pillars, making it difficult to accurately represent distant objects that may only have a few points. Similarly, the range view suffers from the many-to-one problem, which occurs when multiple points in the 3D space are mapped to the exact location in the 2D range image and can cause shape distortions. Therefore, developing more efficient and effective deep learning models for 3D semantic segmentation using point clouds remains an important research direction.

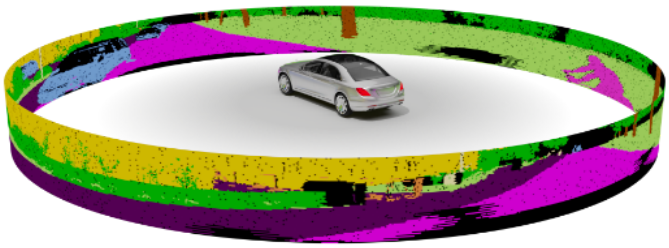


Figure 5.7: Visualization of the range view obtained from a labeled scan in the SemanticKITTI dataset. [32]

The **Range View** representation is chosen as the primary representation in this thesis due to its simplicity and compatibility with widely used convolutional neural networks (CNNs) designed for image semantic segmentation. This representation involves projecting the 3D point cloud data onto a 2D plane based on the range information obtained from the LiDAR sensor. The resulting range view provides a 2D representation of the 3D scene that can be readily processed by CNNs.

To project an individual point n in the point cloud, denoted as $\mathbf{p}_n = (p_n^x, p_n^y, p_n^z)$, onto a 2D image with dimensions $H \times W$, the following mathematical expression [31] is employed:

$$\begin{pmatrix} u_n \\ v_n \end{pmatrix} = \begin{pmatrix} \frac{1}{2}[1 - \arctan(p_n^y, p_n^x)\pi^{-1}]W \\ [1 - (\arcsin(p_n^z, \|\mathbf{p}_n\|^{-1}) + \phi_{down})\xi^{-1}]H \end{pmatrix} \tag{5.1}$$

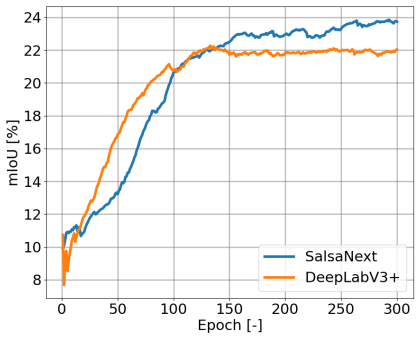
Here, (u_n, v_n) represents the grid coordinate of point p_n in the range image. Additionally, $\xi = |\phi_{up}| + |\phi_{down}|$ represents the vertical field-of-view (FOV) of the sensor, where ϕ_{up} and ϕ_{down} correspond to the inclination angles in the upward and downward directions, respectively.

5.3.2 Model Selection

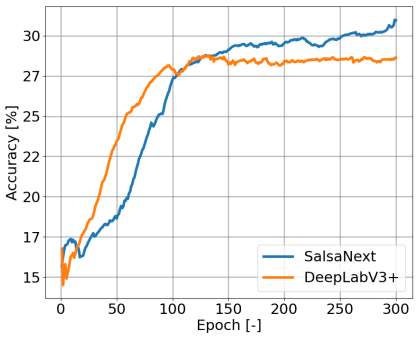
For the purpose of 3D semantic segmentation using point clouds in this thesis, we will focus on comparing two specific architectures: SalsaNext [9] and DeepLabV3+ [33]. SalsaNext is a convolutional neural network (CNN) designed specifically for real-time semantic segmentation of LiDAR scans in the automotive industry. Despite its simplicity, SalsaNext has demonstrated remarkable performance with a relatively low parameter count of 6.73 million. It was the state-of-the-art architecture on the SemanticKITTI

benchmark in 2020. It continues to be one of the leading models for LIDAR semantic segmentation using the range view approach, surpassing models like SqueezeSeg [34].

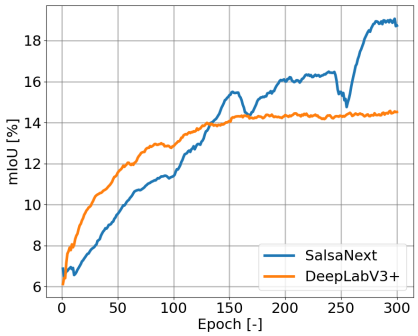
To assess the performance of SalsaNext on our modified datasets 5.2, we will compare it with DeepLabV3+, a well-known architecture widely used for image semantic segmentation. The selected datasets for this comparison are KITTI-360 and SemanticKITTI. We will employ the CrossEntropy loss function to train and evaluate the models and utilize the Adam optimizer with a learning rate of 0.01. By comparing the results of SalsaNext and DeepLabV3+ on these datasets, we aim to gain insights into the strengths and weaknesses of each architecture in the context of 3D semantic segmentation using point clouds.



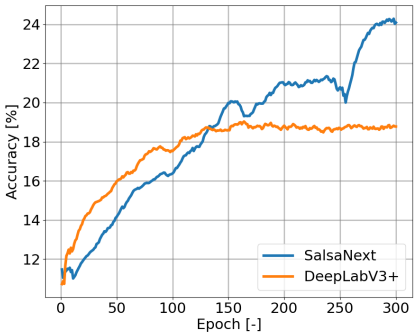
(a) : mIoU on KITTI-360 dataset.



(b) : Accuracy on KITTI-360 dataset.



(c) : mIoU on SemanticKITTI dataset.



(d) : Accuracy on SemanticKITTI dataset

Figure 5.8: Results of SalsaNext and DeepLabV3+ with Cross Entropy Loss.

Results indicate that **SalsaNext** exhibits dominance on our datasets with the aforementioned settings. Therefore, we will use SalsaNext as the example architecture for subsequent experiments. However, it is essential to note that the active learning strategies proposed in this thesis should also apply to other architectures. The focus is on developing active learning approaches that can be generalized and yield similar benefits across various architectures.

5.4 Loss Function

An appropriate loss function is crucial to train a semantic segmentation model effectively [35]. A loss function measures the dissimilarity between the predicted segmentation and the ground truth, providing a signal for the model to optimize its parameters. This section will explore several commonly used loss functions in semantic segmentation and discuss their characteristics and performance.

1. **Cross Entropy Loss:** Cross Entropy Loss is one of the most widely employed loss functions for semantic segmentation. It calculates the pixel-wise cross-entropy between the predicted probability distribution and the ground truth segmentation. Cross Entropy Loss encourages the model to assign high probabilities to correct labels and penalize incorrect predictions. This loss is suitable for balanced datasets with roughly equal class distribution.

2. **Focal Loss:** Focal Loss [36] addresses the issue of class imbalance, which commonly occurs in semantic segmentation. This loss function introduces a modulating factor that downweights the loss for well-classified pixels, emphasizing challenging and misclassified pixels more. By doing so, Focal Loss helps the model focus on learning from complex examples and improves performance on highly imbalanced datasets.

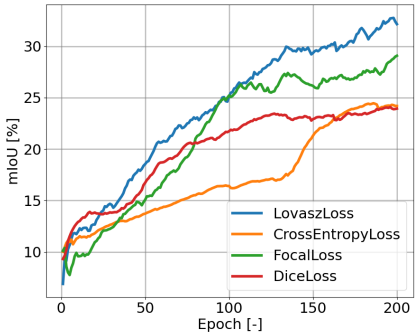
3. **Dice Loss:** Dice Loss [37], also known as the Sørensen-Dice coefficient loss, evaluates the similarity between the predicted segmentation and the ground truth by computing the overlap between their binary masks. It measures the ratio of twice the intersection to the sum of the prediction and ground truth areas. Dice Loss is suitable for datasets with mild class imbalances and performs well when the foreground/background classes have different proportions.

4. **Lovasz-Softmax Loss:** Lovasz Loss [38], also Lovasz Loss is based on submodular losses and provides a continuous relaxation of the intersection-over-union (IoU) measure. It measures the distance between the predicted segmentation and the ground truth by considering the convex loss function of the sorted IoU values. Lovasz Loss is particularly effective when dealing with non-differentiable IoU-based metrics and works well on datasets with various class distributions.

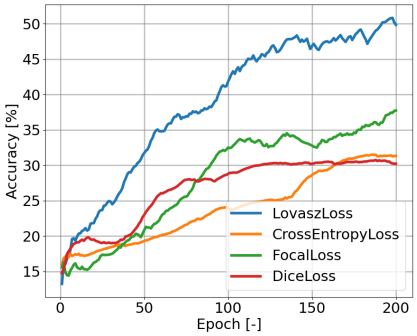
No single loss function performs optimally in all scenarios [35]. The loss function's choice depends on the dataset's characteristics and the segmentation task's specific requirements. Highly imbalanced segmentation tasks tend to benefit from focus-based loss functions. On the other hand, the Cross Entropy loss is more suitable for balanced datasets, while smoothed or generalized dice coefficients can be effective for mildly skewed datasets.

To determine the best loss function for evaluating our active learning method, we conducted experiments using our preprocessed datasets. We

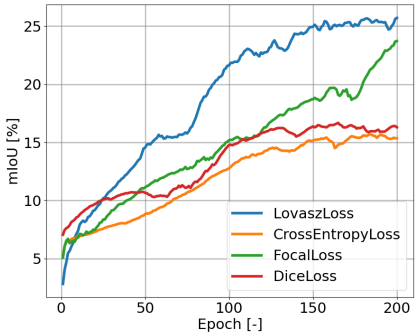
trained the SalsaNext [9] model architecture with different loss functions and evaluated their performance. The results are shown in Figure 5.9.



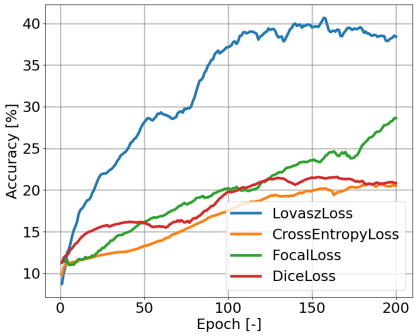
(a) : mIoU on KITTI-360 dataset.



(b) : Accuracy KITTI-360 dataset.



(c) : mIoU on SemanticKITTI dataset.



(d) : Accuracy on SemanticKITTI dataset.

Figure 5.9: Results of different loss functions on KITTI-360 and SemanticKITTI datasets with SalsaNext [9] model architecture.

As shown in Figure 5.9, the SalsaNext architecture with **Lovasz-Softmax** Loss performed the best on our datasets. Therefore, we will utilize this loss function in the subsequent chapter for conducting active learning experiments.

5.5 Data Augmentation

When working with LiDAR (Light Detection and Ranging) data, which is widely used in autonomous driving and 3D perception tasks, augmentations are crucial for expanding the dataset and capturing diverse scenarios. This section discusses four essential data augmentation techniques specifically tailored for LiDAR data: dropping random points, rotating scans, translating points, and flipping points. These techniques aim to increase the diversity and generalization ability of the model, improving its performance across various real-world scenarios.

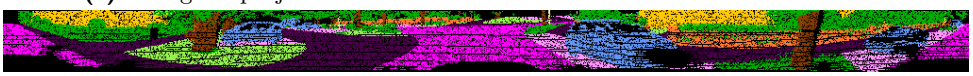
1. **Dropping Random Points in Scan:** The first augmentation technique involves randomly dropping points from the LiDAR scan. This technique introduces sparsity and simulates scenarios where the LiDAR sensor may miss particular objects or encounter occlusion. By removing a percentage of points uniformly or based on a certain distribution, the model learns to handle missing or incomplete data, enhancing its robustness in challenging environments.

2. **Rotating Scan around the Z-Axis:** Rotating the LiDAR scan around the vertical or z-axis is another augmentation technique. This rotation mimics the change in viewpoint or sensor orientation. By applying random rotations within a specific range, the model becomes invariant to the sensor's initial orientation, effectively handling different perspectives and variations in LiDAR data.

3. **Jittering in X, Y, and Z Axes:** LiDAR data augmentation can also involve translating the individual points in the scan along the x, y, and z axes. This technique aims to capture spatial variations and shifts in the environment. Random translations simulate changes in the LiDAR sensor's position, allowing the model to learn robust representations invariant to small spatial displacements.

4. **Flipping Points around the X-Axis:** Flipping points around the x-axis is an augmentation technique commonly used in LiDAR data processing. This transformation can help address biases that may exist due to sensor placement or environmental factors. By randomly mirroring the LiDAR scan, the model learns to handle both left-to-right and right-to-left scenarios, contributing to improved generalization and performance.

(a) : Original projection of the LiDAR scan from SemanticKITTI dataset.



(b) : Projection of the scan where the points were randomly dropped.

(c) : Projection of the scan, which was randomly rotated around the z-axis.

(d) : Projection of the scan with jittering in all axis.

(e) : Projection of the scan that was flipped around the x-axis.

Figure 5.10: Visualization of the different augmentations used on the LiDAR data and their effect on the projected image.

Chapter 6

Experiments

In this chapter, we conduct experiments to evaluate the effectiveness of our active learning method. We follow a systematic evaluation process to compare different selection strategies and assess their performance on the KITTI-360 [21] and SemanticKITTI [22] datasets.

To begin, we train a model using supervised learning on a fully labeled dataset, which serves as our **baseline**. This baseline model is trained with optimal settings, including the choice of the loss function, optimizer, and augmentations. We record the primary metric achieved by the baseline model, typically mIoU (mean Intersection-over-Union) or Pixel Accuracy [1], to establish a performance reference.

Our active learning pipeline operates iteratively, starting with a small randomly selected portion of the dataset (1% in this case). We refer to this initial dataset as the active learning seed. All methods use the same active learning seed and model trained on it, ensuring a consistent starting point.

The active learning pipeline follows a pool-based approach, as illustrated in Figure 4.1. It involves selecting regions in the fused cloud for labeling and subsequent model training on subsets of the dataset. We compare several selection strategies: Confidence (**CONF**) (2.3), Margin (**MAR**) (2.4), Entropy (**ENT**) (2.5), Epistemic Uncertainty (**EPI**) (2.9), **ReDAL** [18], our proposed strategy Viewpoint Variance (**VV**) (4.5) and Random (**RAND**). Additionally, we examine the impact of applying the **DIST** and **NN** filters to these strategies.

It is worth mentioning that although we were unable to replicate the results of the ReDAL method precisely due to various factors such as dataset modifications, differences in model architecture, and variations in the computation of the percentage of labeled points, we made every effort to closely recreate their selection strategy based on their provided code [2]. ReDAL is a cutting-edge selection method that integrates color discontinuity, surface variation, and diversity awareness into the active learning process.

The performance of each selection strategy is compared to the Random selection approach, where regions in the fused cloud are chosen randomly for labeling.

[1] `https://paperswithcode.com/task/semantic-segmentation`

[2] `https://github.com/tsunghan-wu/ReDAL.git`

Next, we will cover the baseline creation on our datasets and then discuss the results and performance analysis of different selection strategies on the KITTI-360 and SemanticKITTI datasets.

6.1 Baseline Training

This section outlines the process of creating the baseline for our experiments. Based on the experiments 5.8 and 5.9 we use the SalsaNext model with the Lovasz-Softmax loss function, Adam optimizer with a learning rate of 0.01, and the augmentations described in Section 5.5.

During the course of our experiments, we realized that evaluating active learning methods in a reasonable time frame required redefining the baseline. Initially, we aimed to train the model until convergence, which experimentally took around 300-400 epochs. However, evaluating active learning methods proved to be more challenging than anticipated. We attempted a checkpoint-based approach, using the weights from the best model of the previous iteration as a starting point and selecting the next set of data for training. However, this approach led to rapid overfitting, as shown in Figure 6.1. The model performed best with only 8% of the labeled voxels, indicating overfitting and suboptimal performance.

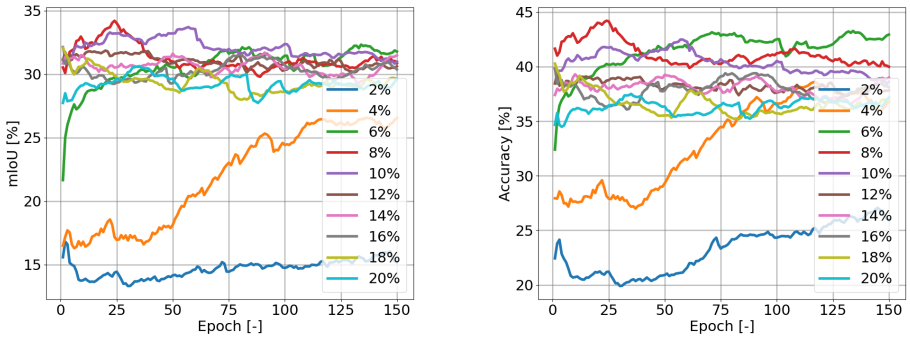


Figure 6.1: An example of overfitting during an active learning experiment. The model performs best with 8% of labeled voxels, indicating overfitting.

To address the issue of overfitting in the checkpoint-based approach, we considered an alternative solution where we would train the model from scratch for each iteration. However, this approach would significantly increase the time required to conduct the experiments. Therefore, as a compromise, we decided to train the model from a checkpoint where overfitting was not observed, which corresponds to the seed model trained on 1% of the data, and reduce the training of the baseline model to the 250 epochs. This compromise allowed us to strike a balance between achieving reliable results and reducing the time complexity of the active learning experiments.

The training can be seen in Figure 6.2. We have been able to train the SalsaNext model to have 43.6% and 31.9% mIoU and 56% and 45.2% Pixel

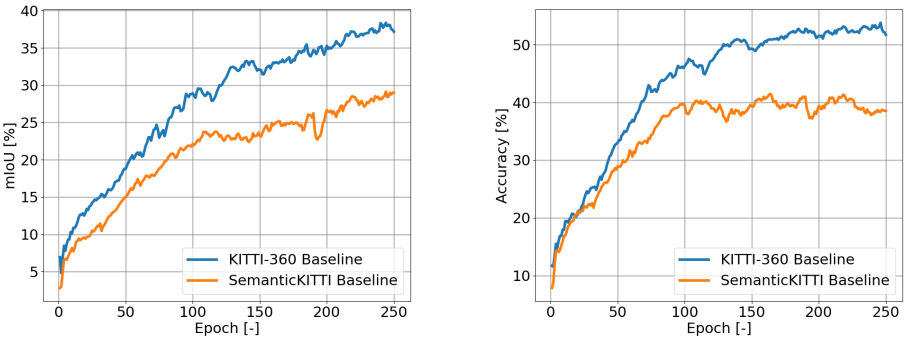


Figure 6.2: Training the SalsaNext model with Lovasz-Softmax loss for 250 epochs as our baseline. The data are distorted by exponential moving average to better visualize the model training trend. The training is limited to 250 epochs to reduce the time complexity of the active learning experiments.

Accuracy on the modified KITTI-360 and the SemanticKITTI respectively. These performances will be our benchmarks for the active learning methods. The 100% of the model's performance on the dataset will be visualized with the dashed line - - - - and the 90% of the performance which will be our target will be represented by the dotted line ·····.

6.2 Performance Analysis on KITTI-360 Dataset

This section presents the results of the experiments conducted on the KITTI-360 dataset. Firstly, we compare the pipeline 4.1 with the proposed uncertainty method and assess the impact of our filters on this proposed framework in contrast to random selection. This evaluation aims to provide an overall assessment of the effectiveness of our methods in comparison to passive learning, which is represented here by random selection. Next, we highlight the disparities between our proposed active learning framework and a random selection, emphasizing the benefits of active learning over passive learning. Subsequently, we compare all uncertainty methods against the state-of-the-art selection strategy, which also incorporates diversity selection criteria, to evaluate our pipeline. Lastly, we conduct a similar comparison but apply filters to all methods to evaluate the effectiveness of the proposed filters.

6.2.1 Comparison to Random Selection

Firstly, we compare the results of our proposed active learning pipeline using the Viewpoint Variance (**VV**) selection strategy (4.5) against the Random (**RAND**) selection of regions. We analyze the Viewpoint Variance method without filtering, with the **DIST** (4.1) filter and the **NN** (4.3) filter. Figure 6.3 illustrates the comparison of these four methods.

The results indicate that all proposed methods outperform random selection at every stage of the active learning experiment. Without any filter, the

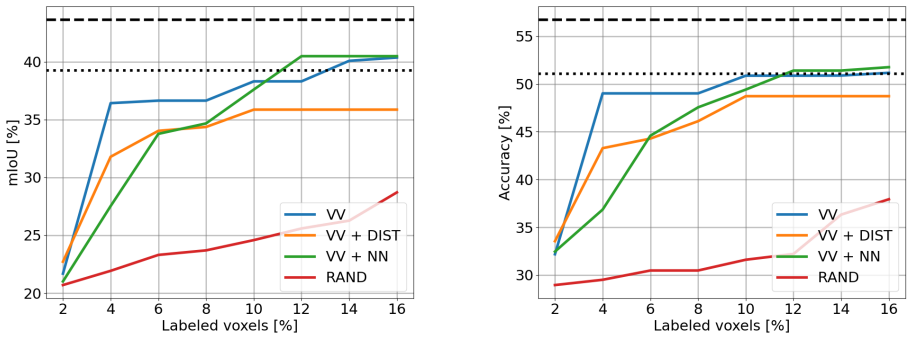
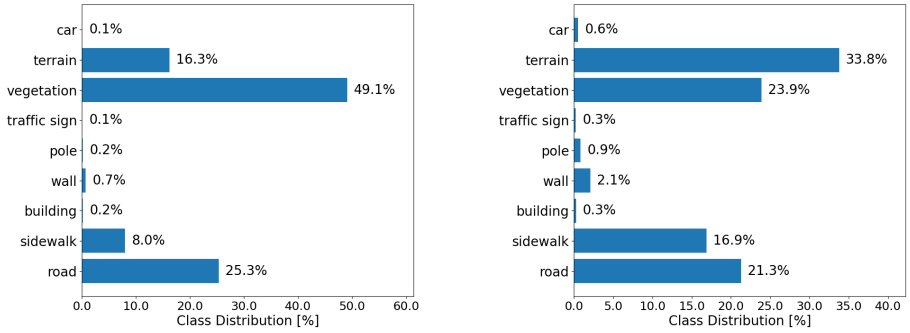


Figure 6.3: Comparison of the proposed uncertainty selection strategy (**VV**) with different point cloud filters against random selection.

VV method achieves the best performance with a limited amount of available data. However, with a larger dataset, the **NN** filter proves to be valuable and surpasses the performance of the method without any filtering. Specifically, the $\mathbf{VV_{NN}}$ method achieves 90% of the baseline performance with only 16% of the available data. On the other hand, the **DIST** filter does not provide any significant improvement at any stage of the active learning experiment.

We will now present the dataset distribution of the classes in order to visualize the difference between our proposed active learning strategy (**VV**) without any filter and the passive learning method, random selection.



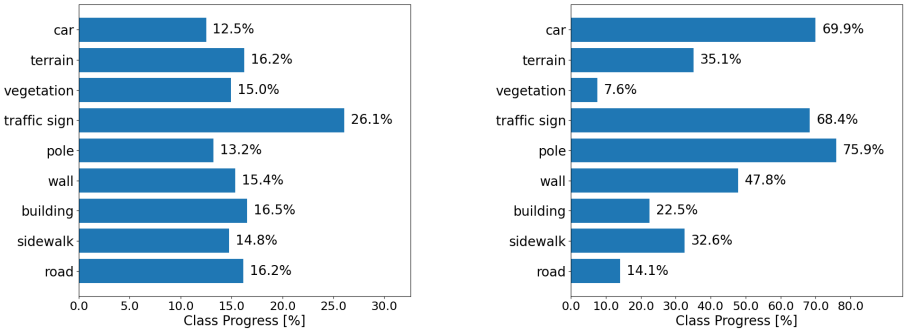
(a) : Labeled dataset class distribution with random selection.

(b) : Labeled dataset class distribution with **VV** strategy.

Figure 6.4: Comparison of class distributions of the labeled datasets between random selection strategy and the selection based on Viewpoint Variance (**VV**) after selecting 16% of the dataset.

As shown in Figure 6.4, we can observe that the dataset selected by the **VV** strategy creates a more balanced dataset.

To further elucidate the distinction between random selection and selection based on the **VV** strategy, we examine the plot shown in Figure 6.5, which illustrates the progress of class labeling.



(a) : Class labeling progress with random selection.

(b) : Class labeling progress with **VV** selection strategy.

Figure 6.5: Comparison of the labeling progress, indicating the preference for each class between the **VV** strategy and random selection.

From the plots, it is evident that the **VV** strategy tends to prioritize the less represented classes, resulting in a more balanced labeled class distribution. We observe that the most represented class, *vegetation*, is the least preferred by the selection strategy. Therefore, the selection process aims to minimize redundancy in the dataset by avoiding overemphasizing the dominant class.

6.2.2 Framework Evaluation

We will now proceed to the comparison of the different selection strategies using our pipeline, which should improve the robustness of the uncertainty strategies. This will be done without the application of any proposed filters.

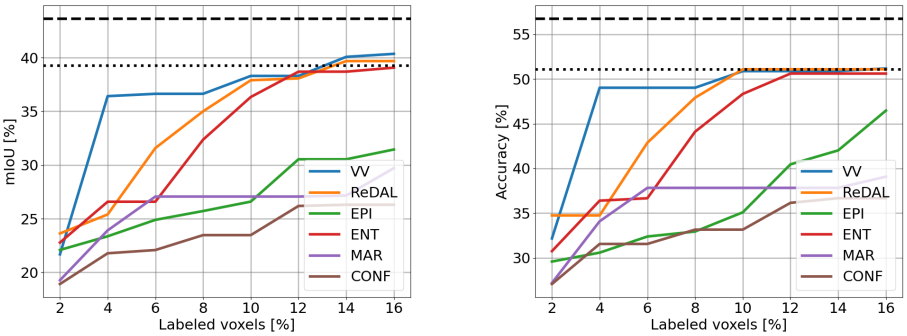


Figure 6.6: Comparison of the different active learning strategies without filters on our pipeline.

Based on this experiment, we can see that the **VV** strategy works best with our framework surpassing the **ReDAL** strategy. The vital thing to note is also that the **ENT** strategy, which performed poorly in comparison to the **ReDAL** in the study [18] or [8] now achieves similar results, indicating possible improvement with our pipeline.

6.2.3 Filter Evaluation

Next, we assess the impact of the **DIST** and **NN** filters by conducting the experiment presented in Figure 6.6 with the filters applied.

First, we examine the impact of the **DIST** filter with the parameter $\rho = 30$, and the results are shown in Figure 6.7.

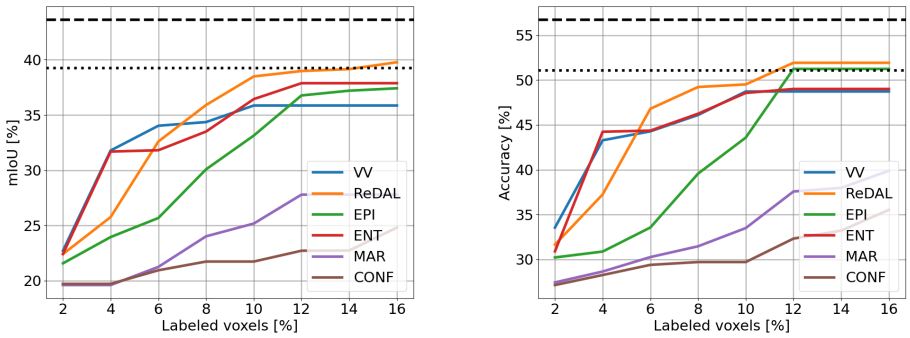


Figure 6.7: Comparison of the different selection strategies with **DIST** filter.

The results show that the **ReDAL** strategy demonstrates improvement with the **DIST** filter, achieving the highest performance in both metrics. It reaches the 90% baseline benchmark with 14% of the dataset labeled.

Similarly, we evaluate the impact of the **NN** filter with the parameters $\varepsilon = 0.1$ and $k = 20$ on the selection strategies, and the results are presented in Figure 6.8.

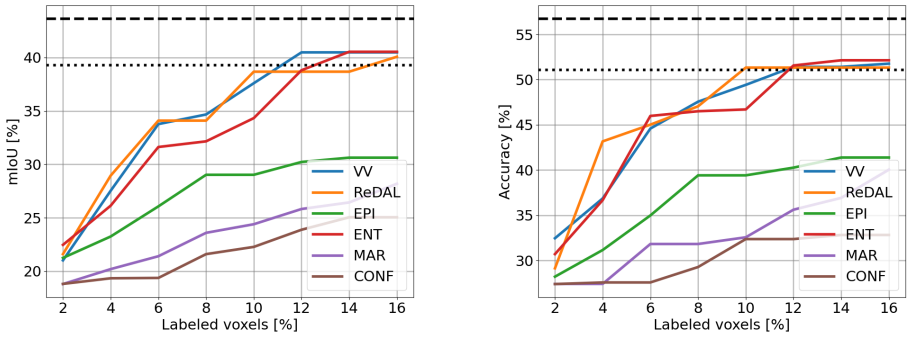


Figure 6.8: Comparison of the different active learning strategies with **NN** filter.

From the results, we observe an improvement in the **ENT** selection strategy, which outperforms **ReDAL** and achieves the 90% baseline benchmark with only 14% of the dataset labeled. The **VV** strategy also demonstrates improvement and reaches the 90% benchmark with 12% of the dataset labeled. Therefore, the $\mathbf{VV_{NN}}$ selection strategy, which combines the **VV** method with the **NN** filter, emerges as the best active learning method on the modified KITTI-360 dataset based on our experiments.

To compare all the experiments, please refer to Table A.1 and Table A.2.

6.3 Performance Analysis on SemanticKITTI Dataset

The analysis of the results obtained from the modified SemanticKITTI dataset follows a similar structure to the analysis conducted on the KITTI-360 dataset in Section 6.2. In this section, we will first demonstrate the effectiveness of our proposed active learning method compared to the passive learning approach, which is random selection. We will then compare all strategies within our pipeline to evaluate its overall performance. Finally, we will repeat the experiment with the applied filters proposed in order to evaluate their impact on the active learning methods.

6.3.1 Comparison to Random Selection

Similar to the experiment conducted on the KITTI-360 dataset (refer to Section 6.2.1), we perform the same experiment on the SemanticKITTI dataset to compare all variations of our proposed method against random selection.

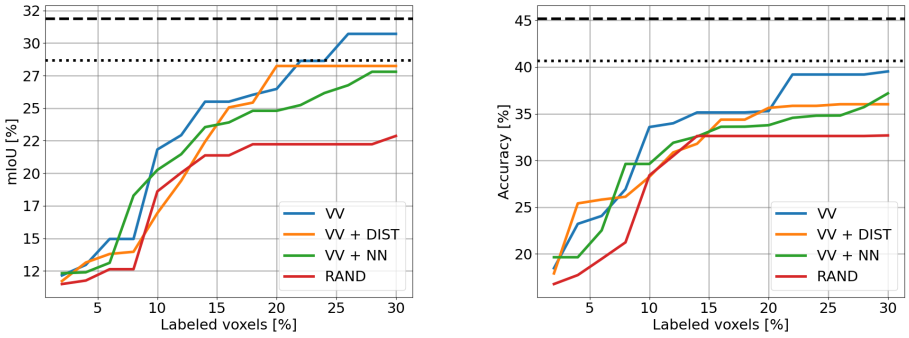


Figure 6.9: Comparison of the proposed uncertainty selection strategy **VV** with different filters to random selection.

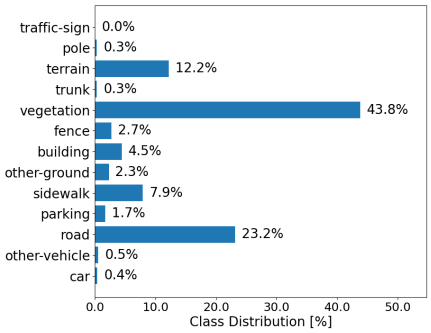
The results demonstrate that, similar to the experiments conducted on the KITTI-360 dataset, all proposed methods outperform random selection in most stages of the active learning experiment. However, it should be noted that the $\mathbf{VV_{NN}}$ variation performs less effectively compared to the other proposed variations of our method. This discrepancy can be attributed to the different parameters used for filtering unreliable points. Nonetheless, as observed in the KITTI-360 experiment (Section 6.2.1), all variations show superior results compared to random selection. Furthermore, the results indicate that the **VV** strategy can achieve the 90% benchmark with only 22% of the dataset available.

To gain further insights into the distinction between random selection and the selection based on the **VV** strategy in the context of the SemanticKITTI

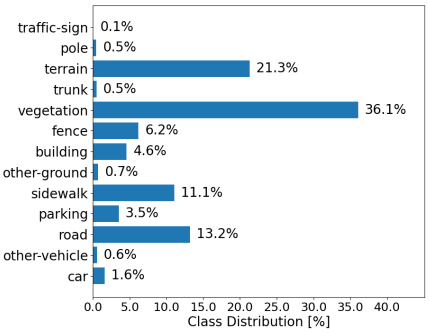
dataset, we examine the plots presented in Figure 6.10. These plots provide statistics regarding the selection preference of each method.

Upon analyzing the plots, it becomes apparent that the **VV** strategy prioritizes the less represented classes, resulting in a more balanced distribution of labeled class instances, similar to the observations made in the previous section's experiment. This balanced class distribution positively impacts the performance of the loss function. Additionally, upon closer examination of the results, we observe that the **VV** strategy specifically avoids overemphasizing the most represented classes, such as *vegetation* and *road*, which are among the three least preferred classes selected by the strategy. This highlights the strategy's ability to minimize redundancy in the dataset and prevent an excessive focus on dominant classes, consistent with the findings from the KITTI-360 experiments.

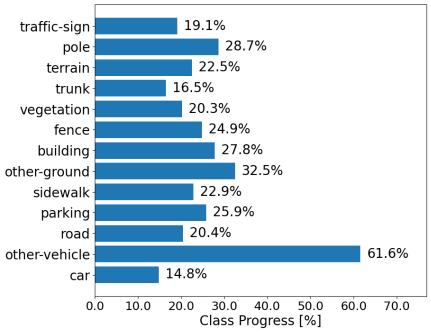
The plots in Figure 6.10 provide visual evidence of the effectiveness of the **VV** strategy in creating a more balanced labeled class distribution, which can contribute to improved overall performance.



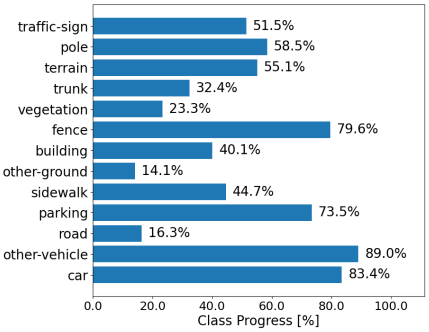
(a) : Labeled dataset class distribution with random selection.



(b) : Labeled dataset class distribution with **VV** strategy.



(c) : Class labeling progress with random selection.



(d) : Class labeling progress with **VV** selection strategy.

Figure 6.10: A comparison between the random selection strategy and the selection based on Viewpoint Variance after selecting 30% of the dataset. The top two plots depict the distribution of labeled classes in the dataset, while the bottom two plots illustrate the labeling progress for each class.

6.3.2 Framework Evaluation

In the evaluation of the framework on the SemanticKITTI dataset, as shown in Figure 6.11, we observe that the proposed Viewpoint Variance (**VV**) strategy consistently performs as one of the best methods across different stages of the active learning experiment, without the application of any filters. This finding is consistent with our previous experiments on the KITTI-360 dataset.

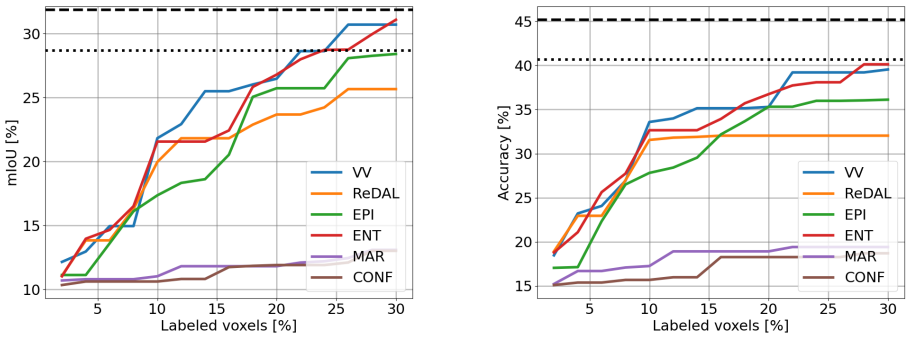


Figure 6.11: Comparison of the different active learning strategies without filters. Here, we can observe that the proposed **VV** strategy consistently performs as one of the best methods across various stages of the experiment.

However, it's worth noting that the Entropy (**ENT**) strategy shows similar performance to the **VV** strategy and eventually surpasses it with the selection of 30% of the dataset. This suggests that the **ENT** strategy effectively captures uncertainty and identifies informative samples.

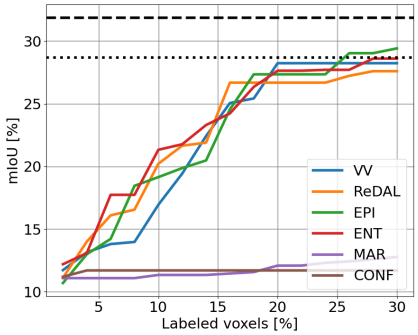
Surprisingly, the Epistemic Uncertainty (**EPI**) strategy exhibits even better performance than the **ReDAL** strategy in this experiment.

Overall, these results demonstrate the effectiveness of the proposed uncertainty-based selection strategies within our framework on the SemanticKITTI dataset, showcasing the potential for improving the active learning process and achieving high-performance results.

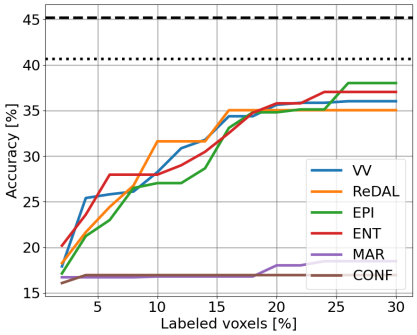
6.3.3 Filter Evaluation

In the evaluation of the **DIST** filter with the parameter $\rho = 30$ and **NN** fillter with the parameters $\varepsilon = 0.1$ and $k = 20$ on the SemanticKITTI dataset, as depicted in Figure 6.12, we observe notable changes in the performance of the active learning strategies.

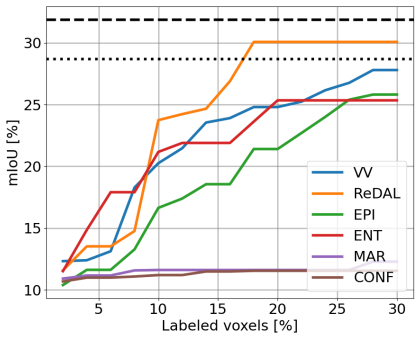
When the **DIST** filter is applied, the performance of most uncertainty-based methods remains comparable to the experiment without filters. However, the **EPI** strategy shows a slight increase in performance with the **DIST** filter, suggesting that the filter might have improved the reliability of uncertainty estimates for this strategy. Additionally, the **ReDAL** strategy also demonstrates an improvement with the **DIST** filter, making the combination of **ReDAL** with the **DIST** filter ($\textbf{ReDAL}_{\textbf{DIST}}$) the best performing method in this experiment.



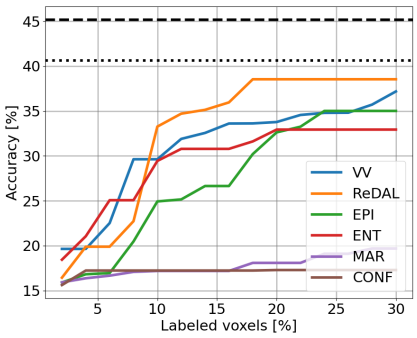
(a) : mIoU with the **DIST** filter.



(b) : Accuracy with the **DIST** filter.



(c) : mIoU with the **NN** filter.



(d) : mIoU with the **NN** filter.

Figure 6.12: Comparison of the different active learning strategies with **DIST** and **NN** filters.

Interestingly, the most significant change is observed with the application of the **NN** filter. All purely uncertainty-based methods show a decrease in performance compared to the experiment without filters, suggesting that the chosen parameters were not suitable for this dataset. It is possible that alternative parameter choices could have led to improved performance for these methods. However, the **ReDAL** strategy shows improvement with the **NN** filter, resulting in the combination of **ReDAL** with the **NN** filter (**ReDAL$_{\mathbf{NN}}$**) being the best method for the SemanticKITTI dataset in this experiment.

It is important to note that the exact reasons behind these performance changes can only be speculated. However, it is possible that the diversity-aware selection employed by the **ReDAL** strategy plays a crucial role in its improved performance with the filters.

For a comprehensive comparison of all the experiments conducted on the SemanticKITTI dataset, please refer to Table A.3 and Table A.4.

Chapter 7
Conclusion

In this thesis, we presented a novel active learning framework for reducing the annotation cost of LiDAR datasets in the context of semantic segmentation. Our framework was tested on two widely used datasets, KITTI-360 and SemanticKITTI, which were modified for our purposes. We were able to achieve 90% of the baseline model's performance, trained on fully labeled datasets, with only 12% and 22% of the data labeled for KITTI-360 and SemanticKITTI, respectively.

Our approach leveraged information from multiple viewpoints to obtain more reliable estimates of the model's uncertainty regarding objects in the scene. We introduced the Viewpoint Variance uncertainty selection strategy, which utilized the variance of model predictions from different viewpoints. This strategy demonstrated comparable performance to the state-of-the-art ReDAL method.

Furthermore, we explored the impact of filtering sparse or distant points in the scene to improve the reliability of the model's uncertainty predictions for data selection. However, the results were inconclusive as we had insufficient time to optimize the filter parameters.

Overall, our proposed framework addresses the challenges of sequence LiDAR-based datasets with multiple viewpoints and enhances uncertainty-based active learning strategies, making them more applicable to real-life annotation scenarios for such datasets.

7.1 Future Work

In future research, we aim to evaluate our method on complete datasets to enable more comprehensive comparisons with other approaches. We also intend to refine the proposed filters to further investigate their effectiveness.

Additionally, we recognize the need for a framework that considers the annotation time for individuals using such a framework. Currently, the selected regions in our thesis, as well as in ReDAL [18], often contain multiple classes, which can be challenging for annotators. Ideally, we would like to develop a framework that selects semantically homogeneous regions, requiring annotators to assign only one class per region, thereby reducing the annotation complexity.

To improve the efficiency of the annotation process, we propose incorporating a neural network that can learn to partition point clouds during the active learning process. This would further reduce the time complexity associated with manual annotation.

By addressing these future research directions, we aim to advance the field of active learning for LiDAR-based dataset annotation and contribute to the development of more practical and efficient frameworks for semantic segmentation.

Bibliography

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database,"

[2] C. Sager, C. Janiesch, and P. Zschech, "A survey of image labelling for computer vision applications," *Journal of Business Analytics*, vol. 4, pp. 91–110, 4 2021.

[3] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 3292–3310, 9 2021.

[4] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss, "Towards 3d lidar-based semantic scene understanding of 3d point cloud sequences: The semantickitti dataset," *International Journal of Robotics Research*, vol. 40, pp. 959–967, 8 2021.

[5] Y. Siddiqui, J. Valentin, and M. Nießner, "Viewal: Active learning with viewpoint entropy for semantic segmentation."

[6] M. S. Supervisor, H.-J. B. Advisor, M. Sundholm, and I.-G. Farcas, "Deep active learning for classification tasks,"

[7] B. Settles, "Active learning literature survey."

[8] N. Samet, O. Siméoni, G. Puy, G. Ponimatkin, R. Marlet, and V. Lepetit, "You never get a second chance to make a good first impression: Seeding active learning for 3d semantic segmentation," 4 2023.

[9] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving."

[10] S. Huang, Y. Xie, S. C. Zhu, and Y. Zhu, "Spatio-temporal self-supervised representation learning for 3d point clouds," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6515–6525, 9 2021.

[11] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra, "Self-supervised pre-training of 3d features on any point-cloud," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 10232–10243, 1 2021.

[12] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Point-contrast: Unsupervised pre-training for 3d point cloud understanding," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12348 LNCS, pp. 574–591, 7 2020.

[13] L. Yi, B. Gong, and T. Funkhouser, "Complete label: A domain adaptation approach to semantic segmentation of lidar point clouds," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 15358–15368, 7 2020.

[14] C. Saltori, F. Galasso, G. Fiameni, N. Sebe, E. Ricci, and F. Poiesi, "Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13693 LNCS, pp. 586–602, 7 2022.

[15] Y. Zhang, Z. Li, Y. Xie, Y. Qu, C. Li, and T. Mei, "Weakly supervised semantic segmentation for large-scale point cloud," *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, vol. 4B, pp. 3421–3429, 12 2022.

[16] M. Cheng, L. Hui, J. Xie, and J. Yang, "Sspc-net: Semi-supervised semantic 3d point cloud segmentation network," *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, vol. 2A, pp. 1140–1147, 4 2021.

[17] Z. Liu, X. Qi, and C. W. Fu, "One thing one click: A self-training approach for weakly supervised 3d semantic segmentation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1726–1736, 4 2021.

[18] T.-H. Wu, Y.-C. Liu, Y.-K. Huang, H.-Y. Lee, H.-T. Su, P.-C. Huang, and W. H. Hsu, "Redal: Region-based and diversity-aware active learning for point cloud semantic segmentation."

[19] X. Shi, X. Xu, K. Chen, L. Cai, C. S. Foo, and K. Jia, "Label-efficient point cloud semantic segmentation: An active learning approach," 1 2021.

[20] F. Shao, Y. Luo, P. Liu, J. Chen, Y. Yang, Y. Lu, and J. Xiao, "Active learning for point cloud semantic segmentation via spatial-structural diversity reasoning," 2 2022.

[21] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 3292–3310, 9 2021.

[22] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss, "Towards 3d lidar-based semantic scene understanding of 3d point cloud sequences: The semantickitti dataset," *International Journal of Robotics Research*, vol. 40, pp. 959–967, 8 2021.

[23] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs."

[24] R. Blomley, B. Jutzi, and M. Weinmann, "Classification of airborne laser scanning data using geometric multi-scale features and different neighbourhood types," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III-3, pp. 169–176, 6 2016.

[25] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Ieee transactions on pattern analysis and machine intelligence 1 deep learning for 3d point clouds: A survey,"

[26] 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," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 11105–11114, 11 2019.

[27] H. Thomas, C. R. Qi, J. E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 6410–6419, 4 2019.

[28] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 3070–3079, 4 2019.

[29] X. Zhu, H. Zhou, T. Wang, F. Hong, W. Li, Y. Ma, H. Li, R. Yang, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar-based perception," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 9 2021.

[30] V. E. Liong, T. N. T. Nguyen, S. Widjaja, D. Sharma, and Z. J. Chong, "Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation," 12 2020.

[31] L. Kong, Y. Liu, R. Chen, Y. Ma, X. Zhu, Y. Li, Y. Hou, Y. Qiao, and Z. Liu, "Rethinking range view representation for lidar segmentation," 3 2023.

[32] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Ieee transactions on pattern analysis and machine intelligence 1 deep learning for 3d point clouds: A survey,"

[33] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11211 LNCS, pp. 833–851, 2 2018.

[34] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1887–1893, 10 2017.

[35] S. Jadon, "A survey of loss functions for semantic segmentation," *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2020*, 6 2020.

[36] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 318–327, 8 2017.

[37] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10553 LNCS, pp. 240–248, 7 2017.

[38] 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," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4413–4421, 5 2017.

Appendix A
Tables

Method	2%	4%	6%	8%	10%	12%	14%	16%
RAND	20.7	21.9	23.3	23.7	24.6	25.6	26.3	28.7
CONF	18.9	21.8	22.1	23.5	23.5	26.2	26.3	26.3
$\text{CONF}_{\text{DIST}}$	19.7	19.7	20.9	21.7	21.7	22.7	22.7	24.8
CONF_{NN}	18.8	19.3	19.4	21.6	22.3	23.9	25.0	25.0
MAR	19.3	23.9	27.1	27.1	27.1	27.1	27.2	29.7
MAR_{DIST}	19.6	19.6	21.2	24.0	25.2	27.8	27.8	27.8
MAR_{NN}	18.8	20.2	21.4	23.6	24.4	25.8	26.4	28.2
EPI	22.1	23.4	24.9	25.7	26.6	30.5	30.5	31.4
EPI_{DIST}	21.6	23.9	25.7	30.1	33.1	36.8	37.2	37.4
EPI_{NN}	21.2	23.2	26.1	29.0	29.0	30.2	30.6	30.6
ENT	22.8	26.6	26.6	32.4	36.3	38.7	38.7	39.1
ENT_{DIST}	22.4	31.7	31.8	33.5	36.4	37.9	37.9	37.9
ENT_{NN}	22.5	26.1	31.6	32.1	34.3	38.8	**40.5**	**40.5**
ReDAL	**23.6**	25.4	31.6	35.0	37.9	38.1	39.7	39.7
$\text{ReDAL}_{\text{DIST}}$	22.4	25.8	32.6	35.9	38.5	39.0	39.2	39.8
ReDAL_{NN}	21.6	28.9	34.1	34.1	**38.7**	38.7	38.7	40.1
VV	21.7	**36.4**	**36.6**	**36.6**	38.3	38.3	40.1	40.4
VV_{DIST}	22.7	31.8	34.0	34.4	35.9	35.9	35.9	35.9
VV_{NN}	21.0	27.5	33.8	34.7	37.6	**40.5**	**40.5**	**40.5**

Table A.1: Comparison of the highest mIoU values achieved by the SalsaNext model at each iteration of the active learning framework, using different selection strategies and filters, on the KITTI-360 dataset. Based on these strategies, the dataset is incrementally expanded by 2% at each iteration.

Method	2%	4%	6%	8%	10%	12%	14%	16%
RAND	28.9	29.5	30.5	30.5	31.6	32.2	36.3	37.9
CONF	27.1	31.6	31.6	33.2	33.2	36.2	36.7	36.7
$\text{CONF}_{\text{DIST}}$	27.1	28.3	29.4	29.7	29.7	32.3	33.2	35.5
CONF_{NN}	27.4	27.6	27.6	29.3	32.4	32.4	32.8	32.8
MAR	27.2	34.1	37.8	37.8	37.8	37.8	37.8	39.1
MAR_{DIST}	27.4	28.6	30.3	31.5	33.5	37.6	38.0	39.9
MAR_{NN}	27.4	27.4	31.8	31.8	32.6	35.6	36.9	40.0
EPI	29.6	30.6	32.4	32.9	35.1	40.5	42.0	46.5
EPI_{DIST}	30.2	30.9	33.6	39.6	43.6	51.2	51.2	51.2
EPI_{NN}	28.2	31.1	35.0	39.4	39.4	40.3	41.4	41.4
ENT	30.7	36.4	36.7	44.1	48.3	50.6	50.6	50.6
ENT_{DIST}	30.9	44.2	44.4	46.2	48.6	49.0	49.0	49.0
ENT_{NN}	30.7	36.7	46.0	46.5	46.7	51.6	**52.1**	**52.1**
ReDAL	**34.7**	34.7	42.9	47.9	51.1	51.1	51.1	51.1
$\text{ReDAL}_{\text{DIST}}$	31.6	37.2	46.8	**49.2**	49.5	**51.9**	51.9	51.9
ReDAL_{NN}	29.1	43.2	45.0	47.0	**51.3**	51.3	51.3	51.3
VV	32.2	**49.0**	**49.0**	49.0	50.9	50.9	50.9	51.2
VV_{DIST}	33.5	43.3	44.3	46.1	48.7	48.7	48.7	48.7
VV_{NN}	32.5	36.9	44.6	47.6	49.4	51.4	51.4	51.8

Table A.2: Comparison of the highest Accuracy values achieved by the SalsaNext model at each iteration of the active learning framework, using different selection strategies and filters, on the KITTI-360 dataset. Based on these strategies, the dataset is incrementally expanded by 2% at each iteration.

Method	2%	4%	6%	8%	10%	12%	14%	16%	18%	20%	22%	24%	26%	28%	30%
RAND	11.5	11.8	12.6	12.6	18.6	20.0	21.4	21.4	22.2	22.2	22.2	22.2	22.2	22.2	22.9
CONF	10.3	10.6	10.6	10.6	10.6	10.8	10.8	11.7	11.8	11.9	11.9	11.9	12.1	13.0	13.0
$\text{CONF}_{\text{DIST}}$	11.2	11.7	11.7	11.7	11.7	11.7	11.7	11.7	11.7	11.7	11.7	11.7	11.7	11.7	11.7
CONF_{NN}	10.7	11.0	11.0	11.1	11.2	11.2	11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5
MAR	10.7	10.8	10.8	10.8	11.0	11.8	11.8	11.8	11.5	11.5	11.5	11.5	11.5	11.5	11.5
MAR_{DIST}	11.1	11.1	11.1	11.1	11.3	11.3	11.3	11.6	11.6	12.1	12.1	12.3	12.4	12.6	12.8
MAR_{NN}	10.9	11.2	11.2	11.6	11.6	11.6	11.6	11.6	11.6	11.6	11.6	11.6	11.6	12.3	12.3
EPI	11.1	11.1	13.6	16.1	17.4	18.3	18.6	20.5	25.1	25.7	25.7	25.7	28.1	28.3	28.4
EPI_{DIST}	10.7	13.0	14.2	18.5	19.1	19.9	20.5	24.6	27.4	27.4	27.4	27.4	29.0	29.0	29.4
EPI_{NN}	10.4	11.6	11.6	13.3	16.6	17.4	18.6	18.6	21.4	21.4	22.7	24.0	25.4	25.8	25.8
ENT	11.0	14.0	14.7	16.5	21.6	21.6	21.6	22.4	25.8	26.8	28.0	28.7	28.8	30.0	**31.1**
ENT_{DIST}	12.2	13.1	17.7	17.7	21.3	21.8	23.3	24.2	26.3	27.6	27.6	27.7	27.7	28.6	28.6
ENT_{NN}	11.5	**14.9**	**17.9**	**17.9**	21.2	21.9	21.9	21.9	23.6	25.3	25.3	25.3	25.3	25.3	25.3
ReDAL	11.1	13.8	13.8	16.3	20.0	21.8	21.8	21.8	22.9	23.7	23.7	24.2	25.7	25.7	25.7
$\text{ReDAL}_{\text{DIST}}$	11.1	14.0	16.1	16.6	20.2	21.7	21.9	26.7	26.7	26.7	26.7	26.7	27.2	27.6	27.6
ReDAL_{NN}	11.6	13.5	13.5	14.8	**23.7**	**24.2**	24.7	**26.9**	**30.1**	**30.1**	**30.1**	**30.1**	30.1	30.1	30.1
VV	12.2	13.0	15.0	15.0	21.8	22.9	**25.5**	25.5	26.0	26.5	28.6	28.6	**30.7**	**30.7**	30.7
VV_{DIST}	11.7	13.1	13.8	14.0	16.9	19.4	22.4	25.1	25.4	28.2	28.2	28.2	28.2	28.2	28.2
VV_{NN}	**12.3**	12.4	13.1	18.3	20.3	21.5	23.5	23.9	24.8	24.8	25.2	26.2	26.8	27.8	27.8

Table A.3: Comparison of the highest mIoU values achieved by the SalsaNext model at each iteration of the active learning framework, using different selection strategies and filters, on the SemanticKITTI dataset. Based on these strategies, the dataset is incrementally expanded by 2% at each iteration.

Method	2%	4%	6%	8%	10%	12%	14%	16%	18%	20%	22%	24%	26%	28%	30%
RAND	16.8	17.7	19.5	21.2	28.4	30.5	32.6	32.6	32.6	32.6	32.6	32.6	32.6	32.6	32.7
CONF	15.1	15.4	15.4	15.7	15.7	16.0	16.0	18.3	18.3	18.3	18.3	18.3	18.3	18.7	18.7
$\text{CONF}_{\text{DIST}}$	16.1	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
CONF_{NN}	15.6	17.2	17.2	17.2	17.2	17.2	17.2	17.2	17.2	17.3	17.3	17.3	17.3	17.3	17.3
MAR	15.2	16.7	16.7	17.1	17.3	18.9	18.9	18.9	18.9	18.9	19.4	19.4	19.4	19.4	19.4
MAR_{DIST}	16.7	16.7	16.7	16.7	16.8	16.8	16.8	16.8	16.8	18.0	18.0	18.5	18.5	18.5	18.5
MAR_{NN}	15.9	16.4	16.7	17.1	17.2	17.2	17.2	17.2	18.1	18.1	18.1	19.1	19.1	19.7	19.7
EPI	17.0	17.1	22.3	26.5	27.8	28.4	29.5	32.2	33.7	35.3	35.3	36.0	36.0	36.0	36.1
EPI_{DIST}	17.1	21.2	23.0	26.5	27.0	27.0	28.7	33.1	34.8	34.8	35.1	35.1	38.0	38.0	38.0
EPI_{NN}	15.9	16.8	17.0	20.5	24.9	25.2	26.7	26.7	30.2	32.6	33.3	35.0	35.0	35.0	35.0
ENT	18.8	21.1	25.6	27.7	32.6	32.6	32.6	33.9	35.7	36.7	37.7	38.1	38.1	**40.1**	**40.1**
ENT_{DIST}	**20.2**	23.6	**28.0**	28.0	28.0	29.0	30.5	32.5	34.7	35.8	35.8	37.0	37.0	37.0	37.0
ENT_{NN}	18.4	21.1	25.1	25.1	29.4	30.8	30.8	30.8	31.6	32.9	32.9	32.9	32.9	32.9	32.9
ReDAL	18.9	22.9	22.9	27.0	31.5	31.8	31.9	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0
$\text{ReDAL}_{\text{DIST}}$	18.3	21.7	24.4	26.8	31.6	31.6	31.6	35.0	35.0	35.0	35.0	35.0	35.0	35.0	35.0
ReDAL_{NN}	16.4	19.9	19.9	22.7	33.3	**34.7**	**35.1**	**36.0**	**38.5**	**38.5**	38.5	38.5	38.5	38.5	38.5
VV	18.5	23.2	24.1	26.9	**33.6**	34.0	**35.1**	35.1	35.1	35.3	**39.2**	**39.2**	**39.2**	39.2	39.5
VV_{DIST}	17.9	**25.4**	25.8	26.1	28.2	30.9	31.8	34.4	34.4	35.6	35.8	35.8	36.0	36.0	36.0
VV_{NN}	19.6	19.6	22.5	**29.6**	29.6	31.9	32.6	33.6	33.6	33.8	34.6	34.8	34.8	35.7	37.2

Table A.4: Comparison of the highest Accuracy values achieved by the SalsaNext model at each iteration of the active learning framework, using different selection strategies and filters, on the KITTI-360 dataset. Based on these strategies, the dataset is incrementally expanded by 2% at each iteration.

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Kučera Aleš** Personal ID number: **498925**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Active Learning for Semantic Segmentation of Point Clouds

Bachelor's thesis title in Czech:

Aktivní učení pro sémantickou segmentaci mračen bodů

Guidelines:

The aim of the project is the optimal selection of point cloud data samples for the training of semantic segmentation models in the sense of achieving greater accuracy with fewer training labels.
Method:
(a) Select model architecture that takes as input point cloud data and provides semantic labels for each point of the cloud. Proposal: The input point cloud data could be converted to range images. 2D CNN (Deeplab V3 [4]) could be used, which takes as input range images and produces semantic labels for each pixel.
(b) Select a data set to train the model. Proposal: SemanticKITTI [5] or SemanticUSL [6] (datasets have the same format).
(c) Train the model and compute performance metrics, Per-pixel Accuracy and mean Intersection over Union on the validation part of the data set.
(d) Implement an active data samples selection strategy from the same dataset to train the semantic segmentation model, which achieves a similar to baseline performance and requires fewer training samples. Proposal: Use the uncertainty sampling methods to query part of the data set (based on model confidence, entropy, query-by-committee, and margin sampling [1]).
(e) Train the model using the active learning strategy and report achieved performance and the required number of training samples. Compare to the baseline training method (selection of training samples randomly).
(f) Use localisation data to provide consistent predictions for objects in the same environment observed from different positions.

Bibliography / sources:

[1] B. Settles, Active Learning Literature Survey, CS Technical Report, https://burrsettles.com/pub/settles.activelearning.pdf
[2] J. Prendki, An Introduction to Active Learning, ODSC, https://opendatascience.com/an-introduction-to-active-learning/
[3] Active Learning Tutorial, https://towardsdatascience.com/active-learning-tutorial-57c3398e34d
[4] L. Chen et al, Rethinking Atrous Convolution for Semantic Image Segmentation, https://arxiv.org/abs/1706.05587v3
[5] J. Behley et al, Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset, http://semantic-kitti.org/
[6] P. Jiang et al, LiDARNet: A Boundary-Aware Domain Adaptation Model for Lidar Point Cloud Semantic, http://www.unmannedlab.org/research/SemanticUSL

Name and workplace of bachelor's thesis supervisor:

MSc. Ruslan Agishev Vision for Robotics and Autonomous Systems FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **02.01.2023** Deadline for bachelor thesis submission: **26.05.2023**

Assignment valid until: **22.09.2024**

MSc. Ruslan Agishev Supervisor's signature	prof. Ing. Tomáš Svoboda, Ph.D. Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. _____
Date of assignment receipt

_____
Student's signature