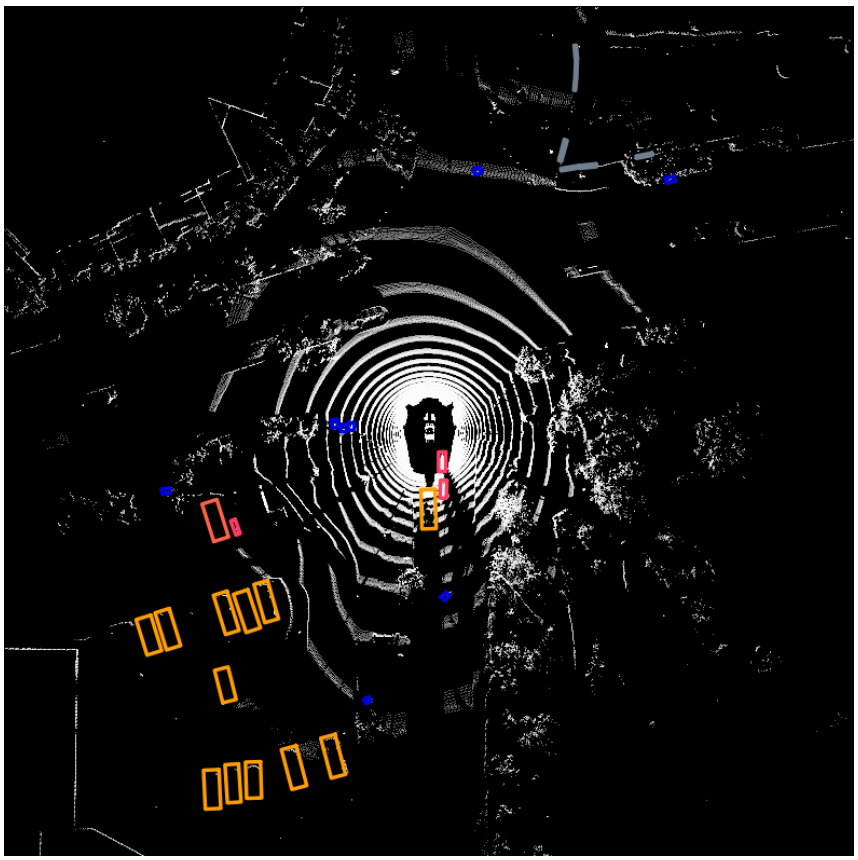
Active VisLED: Active Vision-Language Embedded Diversity Querying for 3D Object Detection



P10 Thesis
Bjørk Antoniussen
Computer Engineering, AI Vision and Sound
Aalborg University
May 30, 2024

AALBORG UNIVERSITY
STUDENT REPORT

Masters Student in Computer Engineering: AI Vision and Sound
Study Board for Electronic Systems
Fredrik Bajers Vej 7
9220 Aalborg

Project:

Master's Thesis

Project Period:

February 2024 - June 2024

Participant:

Bjørk Antoniussen

AAU Supervisor:

Andreas Møgelmose

External Organization Supervisor:

Mohan M. Trivedi

Ross Greer

Page Count: 62
Page Count Incl. Appendix: 75

Abstract:

This thesis presents the development and evaluation of the VisLED-Querying method for 3D object detection, with a particular focus on applications in autonomous driving. By leveraging Vision-Language Embedding Diversity Querying (VisLED-Querying), the study aims to achieve detection performance equivalent to using the full training set, while using only up to 50 % of the dataset, hence reducing the need for extensive labeling. The VisLED-Querying method integrates active learning strategies to select diverse and informative data samples from an unlabeled pool, thereby improving the model's ability to detect underrepresented or novel objects. This approach is evaluated in two scenarios: Open-World Exploring (OWE) and Closed-World Mining (CWM).

Using the nuScenes dataset, the study shows that VisLED-Querying achieves high performance with significantly reduced data. The method reaches performance levels close to a full dataset, even with only 50 % of the data pool. This demonstrates VisLED-Querying's potential to reduce labeling costs and enhance model efficiency, making it valuable for real-world autonomous driving systems. The findings indicate that diversity-based active learning methods, like VisLED-Querying, can lead to more accurate and cost-effective 3D object detection models, advancing autonomous vehicle technologies and other domains requiring robust object detection.

Preface

Acknowledgement

Sincere gratitude is extended to Professor Mohan M. Trivedi for extending my stay and allowing me to continue as a member of the laboratory for my thesis semester, and to Ross Greer for supervising and guiding my project.

Reading

All figures, tables and equations are named sequentially according to the chapter they are in. The figures in Chapter 2 will be *Figure 2.1*, *Figure 2.2* and so forth.

Bjørk Antoniussen
banton19@student.aau.dk

Contents

Summary 1

This thesis investigates the effectiveness of the VisLED-Querying method for 3D object detection, particularly in the context of autonomous driving. The main objective is to achieve equivalent model performance to the utilization of the full dataset, while reducing the amount of labeled data required, thereby minimizing the associated costs and effort. The research addresses a critical challenge in the field: the high cost and time consumption of annotating large datasets, which is necessary for training robust 3D object detection models.

VisLED-Querying is introduced as an innovative active learning (AL) approach that leverages feature embeddings of images to select the most informative and diverse samples for model training. The methodology is designed to maximize learning efficiency and improve model accuracy with fewer, but more diverse labeled data points. Two specific scenarios are examined in this study: Open-World Exploring (OWE) and Closed-World Mining (CWM). Both scenarios utilize the CLIP (Contrastive Language-Image Pre-Training) model to embed images and uncover patterns within the data, facilitating a more nuanced understanding of the dataset.

In the Open-World Exploring scenario, the algorithm focuses on identifying diverse and representative samples from a vast, unlabeled data pool. In the Closed-World Mining scenario, the algorithm prioritizes refining and enhancing the model's performance on a more controlled and predefined set of classes.

Experimental results conducted using the nuScenes dataset, a comprehensive dataset widely used in autonomous driving research, demonstrated that VisLED-Querying significantly outperforms traditional Random Sampling methods. Remarkably, with only 50 % of the data pool, VisLED-Querying achieves performance levels comparable to those obtained with the full dataset. This finding highlights the efficiency of AL approaches in reducing labeling costs while maintaining high model performance. The increased diversity of the sampled data allows the model to learn from unique instances, avoiding the pitfalls of repetitive examples that can lead to confusion and overfitting.

By demonstrating the potential to achieve high model performance with significantly reduced labeled data, this thesis contributes to the ongoing efforts to make autonomous driving technologies more cost-effective and accessible. The findings from this study will hopefully inspire further research and development in AL methodologies, with the aim of improving the efficiency and effectiveness of 3D object detection systems in various real-world applications.

Introduction 2

Over the past several years, there has been a substantial increase in the commitment of automobile manufacturers to the pursuit of autonomous driving. However, despite these efforts and technological advancements, complete vehicle automation has not yet been achieved. Instead, a variety of advanced driver assistance systems (ADAS) are offered, to assist the driver. Making driving safer and alleviating some driver responsibilities.

The progression of vehicle automation has been categorized into six different levels by SAE International in 2021 [1]:

- Level 0: No Driving Automation - The driver oversees all driving tasks with minimal automated assistance.
- Level 1: Driver Assistance - The vehicle incorporates isolated automated features, such as assisted steering or acceleration, but primarily relies on the driver's control.
- Level 2: Partial Driving Automation - The vehicle can handle a combination of functions like steering and acceleration simultaneously, yet requires the driver to remain actively engaged and monitor the surroundings continuously.
- Level 3: Conditional Driving Automation - The vehicle can conduct most driving tasks, although the driver might need to intervene upon the system's request.
- Level 4: High Driving Automation - The vehicle can autonomously perform all driving tasks within specific conditions.
- Level 5: Full Driving Automation - The vehicle can handle all driving operations under any condition, requiring no human intervention.

The National Highway Traffic Safety Admission (NHTSA) has found that 94% of significant accidents are caused by human errors, this is a number that potentially can be significantly decreased by introducing autonomous components into everyday vehicles. Consequently, developing precise and reliable 3D perception models becomes increasingly important, as these models form the backbone of autonomous systems, making it possible to navigate safely in complex environments while creating a better understanding of the obstacles present [2].

Perception systems are a crucial part of autonomous systems, they are responsible for accurately understanding and interpreting the vehicle's surroundings, an important task for decision-making and safe navigation. These systems highly depend on advanced sensors and algorithms for detecting and classifying objects, locating potential dangers, and making informed real-time decisions. As part of the many components of a perception system, an important aspect is 3D object detection. It locates and identifies objects in a driving environment in a three-dimensional space, this is important for creating an increased understanding of the vehicle's surroundings. 3D object detection can enable autonomous

vehicles to predict the movement of surrounding vehicles, accurately measure distances, and understand the spatial relationship of objects. These aspects are crucial when ensuring reliability and safety in autonomous systems. Therefore, this thesis will focus on the 3D object detection task, focusing on LiDAR and Camera sensor fusion-based systems and optimization of data usage techniques [3].

Through recent works, it has been proven that combining the information gathered from the camera and LiDAR sensors is greatly beneficial in the 3D object detection task and creates more precise models. However, this is a challenging task as it relies on a high amount of data sampled and labeled for each sensor [4, 5]. The nuScenes dataset, contains 1.4 million images and 400k LiDAR sweeps which contain 1.4 million labeled objects [6]. According to the RGB-D benchmark for labeling 3D bounding boxes, each object takes 100 seconds to annotate [7], meaning that it would take 38.889 hours to label all objects in the dataset. The annotation of this many objects is both time and cost consuming. This is supported by a German study on autonomous vehicle data, where the cost of annotating datasets is estimated to range from 1.16 trillion to 51.8 trillion Euros per year, which is 14,800 times Germany's gross domestic product [8].

During the annotation phase, a significant challenge emerged due to this annotation bottleneck. It's widely accepted that more data typically leads to better performance. However, a lot of datasets are cluttered with repetitive information, like numerous images/frames showcasing the same vehicle. In an era where data rapidly accumulates, it's vital to reduce redundancy as much as possible before completing the annotation process. The efficiency and effectiveness of annotating data can be greatly enhanced by incorporating a learning algorithm, with active learning (AL) playing a crucial role in this context. AL introduces a proactive method for selecting which data to annotate and utilize for training purposes.

AL is a subset of machine learning where the algorithm selectively queries the user or an oracle to annotate data points with the highest expected utility. This method stands in contrast to the traditional passive learning paradigms, where the learning algorithm is trained on a randomly selected subset of data without any input on the utility of different data points. The fundamental premise of AL is that an algorithm can achieve higher accuracy with fewer training samples if it is allowed to choose the data from which it learns. This not only optimizes resource utilization but also accelerates the path to model maturity by focusing on the diversity of data rather than its volume [9].

Among the various strategies that underpin AL, diversity-based methods and uncertainty-based methods have emerged as the most prevalent and impactful. These methodologies play a pivotal role in optimizing the learning process, ensuring that the selected data for annotation offer the maximum value to the learning algorithm [9].

Uncertainty-based methods prioritize data points for which the model has the lowest confidence in its predictions. These methods leverage the model's own uncertainty to identify the most informative samples. The intuition behind this approach is that by learning from instances where it is least certain, the model can achieve significant improvements in performance with fewer labeled examples. Uncertainty can be measured in various ways, such as the margin between the first and second most probable predictions or the entropy

across all possible outcomes. This approach ensures that the learning algorithm focuses on the most challenging and informative parts of the data, thus accelerating the learning process and enhancing model accuracy in a more targeted manner [9].

While, diversity-based methods focus on selecting a set of data points that are representative of the entire dataset's variability. The core premise is to capture the broad spectrum of data diversity, ensuring that the learning model is exposed to the wide range of examples found within the dataset. This approach helps in building a more generalizable model capable of performing well across diverse scenarios. By prioritizing diversity, these methods aim to reduce the redundancy in the training data, which is especially crucial in datasets with a high degree of similarity among data points. Diversity-based AL is particularly beneficial in scenarios where the goal is to achieve broad coverage of the input space with as few labeled instances as possible, thus maximizing the efficiency of the annotation effort [9].

This research will focus on diversity-based methods due to their lower computational demands compared to uncertainty-based methods, and their significant potential when integrated with new and innovative learning approaches.

Related Works 3

3.1 3D Object Detection

3D object detection is a critical technology with wide-ranging applications, especially in the realm of autonomous driving. Accurately perceiving the environment in three dimensions is fundamental for the safety and navigation capabilities of autonomous vehicles. The evolution of sensor technologies and advancements in machine learning have greatly advanced this field, offering increasingly sophisticated tools for interpreting complex spatial data.

Despite these advancements, the field faces ongoing challenges, particularly in the efficient acquisition and annotation of 3D data, which are essential for training and refining detection algorithms. Moreover, the integration of multi-modal models, which leverage multiple types of sensory data, has proven to enhance the robustness and accuracy of 3D object detection systems. These models can more effectively interpret the rich and varied data types typical in real-world environments, leading to improved detection performance.

This section will explore various architectures that have been developed to address these challenges in 3D object detection. By examining different approaches and their applications, particularly in scenarios demanding high accuracy and reliability, we can better understand the current landscape and future directions of this essential technology. Figure 3.1 presents a comparative overview of a 3D object detection pipeline alongside its 2D counterpart, illustrating the key differences and similarities. [10]

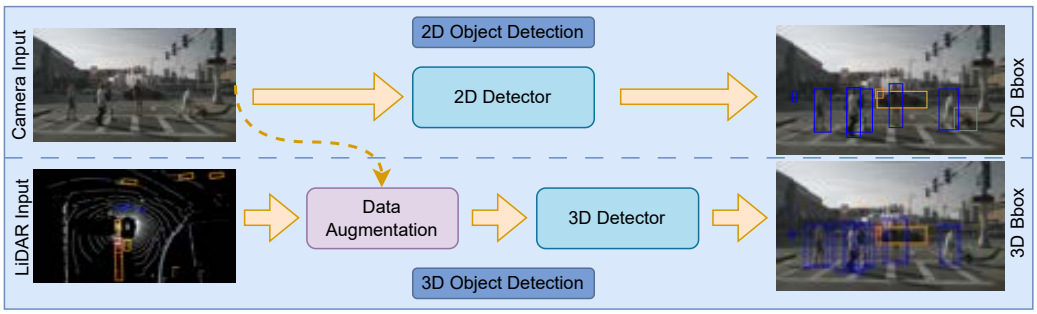


Figure 3.1. Illustration of the difference between 2D object detection and 3D object detection.

In the rapidly evolving field of autonomous driving, effectively integrating data from diverse sensor modalities is crucial for developing reliable and safe navigation systems. Traditional fusion techniques, such as projecting LiDAR data onto camera images or vice versa, encounter significant challenges. These methods often lead to geometric distortion,

shown in Figure 3.2a or semantic dilution, seen in Figure 3.2b, undermining the utility of the fused data for essential tasks like 3D object detection and environmental segmentation.

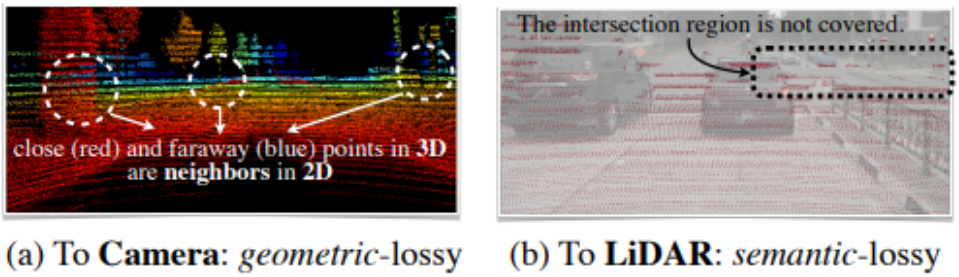


Figure 3.2. Camera-to-LiDAR(a) and LiDAR-to-Camera(b) loss illustration. [4]

3.1.1 Multi-Modal Models

Autonomous vehicles operate in complex environments that require rapid and precise decision-making based on a comprehensive perception of the surroundings. This necessitates the integration of data from multiple sensory inputs, as no single sensor type is adequate in all scenarios. Cameras provide rich, high-resolution color and texture information but are limited by poor performance in low-light conditions and do not provide direct distance measurements. Conversely, LiDAR sensors excel in precise depth sensing and are unaffected by lighting variations but do not capture the semantic details present in visual images [4, 5].

Therefore, the fusion of camera and LiDAR data emerges as a potent solution, merging the strengths of each sensor type to foster a more detailed and resilient understanding of the vehicle's environment. This multi-modal approach enhances the vehicle's ability to detect and interpret objects and obstacles accurately, supporting more robust decision-making processes essential for safe autonomous driving. By combining the high-resolution textural information from cameras with the accurate depth data from LiDAR, autonomous systems can achieve superior object detection capabilities that are crucial in diverse driving conditions.

A prominent architecture is the multi-modal TransFusion architecture [5], the paper introduces a robust framework, seen in Figure 3.3 for LiDAR and camera fusion tailored for 3D object detection. This method operates on a two-stage pipeline: the first stage generates proposals using LiDAR features, and the second stage refines these proposals by fusing them with camera features through transformer-based technology. This approach leverages the complementary strengths of both sensor types to enhance detection accuracy. TransFusion excels in its ability to enhance LiDAR-based proposals with detailed semantic information from camera data, resulting in highly accurate 3D object detection. This fusion strategy significantly improves the precision of object localization and classification by incorporating rich texture and contextual details from camera images into the predominantly geometric data from LiDAR. The method demonstrates substantial improvements in benchmark performance, particularly in complex urban environments where diverse object interactions and varied lighting conditions are common. While TransFusion achieves high accuracy, its two-stage fusion process can introduce additional computational complexity and latency, potentially affecting real-time performance in autonomous driving applications.

Furthermore, the dependency on initial LiDAR-based proposals means that the overall system efficacy might degrade in scenarios where LiDAR data quality is compromised (e.g., adverse weather conditions). Also, the method requires careful alignment and synchronization between LiDAR and camera data, which can be challenging to maintain consistently across different operational conditions.

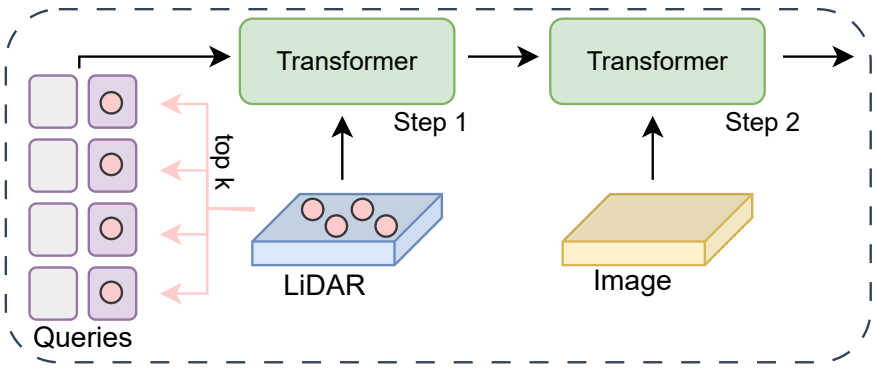


Figure 3.3. Transfusion simple architecture.

The Cross Modal Transformer (CMT) [11], which introduces a novel approach to 3D object detection by directly integrating image and point cloud data without traditional view transformations. Utilizing a transformer-based architecture, CMT enhances the interaction between different modal tokens, achieving impressive performance improvements. CMT's main strength lies in its simple yet effective end-to-end design, seen in Figure 3.4, which facilitates fast and robust 3D object detection. By encoding 3D positional information into multi-modal tokens, it avoids the biases introduced by explicit cross-view feature alignment, simplifying the model architecture.

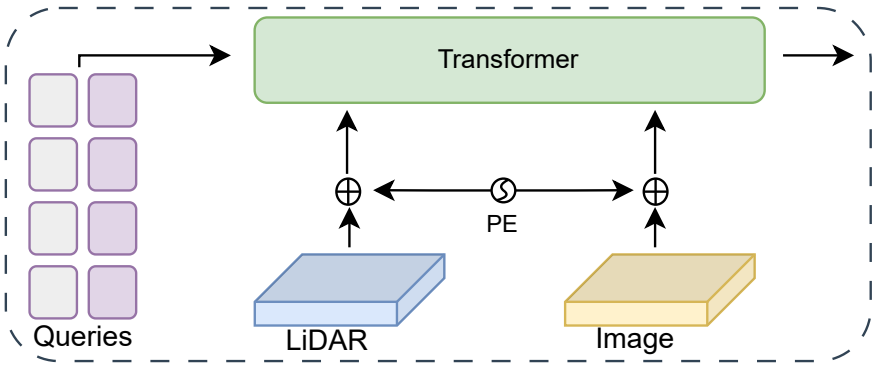


Figure 3.4. Cross Modal Transformer simple architecture.

BEVFusion proposed by Zhijian Liu et al. [4], proposes an efficient and generic multi-task multi-sensor fusion framework, seen in Figure 3.5, which uniquely integrates camera and LiDAR data in a shared bird's-eye view (BEV) representation space. This method maintains both the geometric structure from LiDAR and the semantic density from camera features, effectively supporting various 3D perception tasks including object detection and BEV map

segmentation. BEVFusion significantly enhances the state of the art in 3D object detection, achieving top leaderboard positions on the nuScenes benchmark with improvements in both mAP and NDS metrics [12]. It also demonstrates major advancements in BEV map segmentation, outperforming existing camera-only and LiDAR-only models. The unified BEV approach not only preserves the integrity of input modalities but also reduces computational costs by over 40 times, thanks to optimized BEV pooling operations. The framework, while powerful, requires careful tuning and integration of sensor inputs to maintain performance across varied operational conditions.

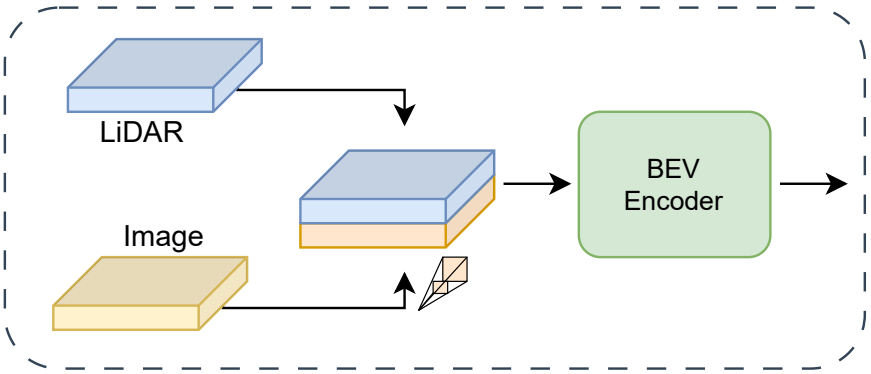


Figure 3.5. Birds-Eye-View (BEV) Fusion simple architecture.

Overall, sensor fusion not only compensates for the individual weaknesses of each sensing modality but also collectively enhances the vehicle's perceptual accuracy, leading to improved navigation and safety outcomes.

3.2 Datasets

The development and testing of autonomous driving technologies depend heavily on diverse and comprehensive datasets. This section reviews several key datasets, noting their contributions and limitations within the context of urban navigation and broader autonomous vehicle research.

3.2.1 KITTI

The KITTI dataset by Geiger et al. [13] is foundational in the field, supporting a variety of tasks such as stereo vision, optical flow, visual odometry, and 3D object detection. Its strengths lie in its diverse real-world driving scenarios and comprehensive sensor suite. However, its primary limitation is the scope of its environments, primarily captured in rural and highway settings, which may not fully represent the complexity of more dynamic urban landscapes.

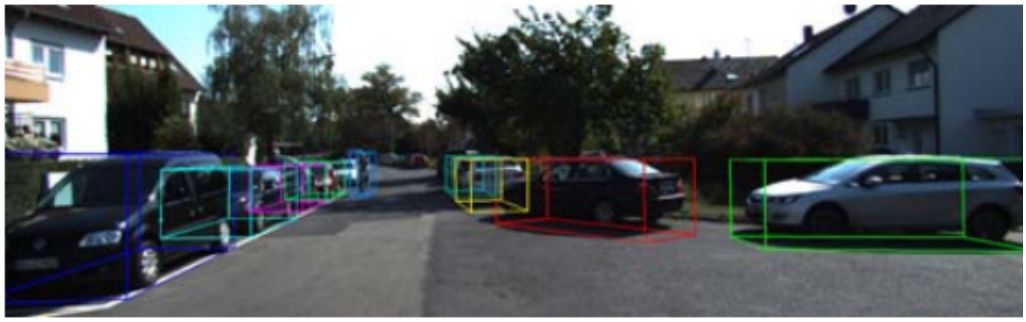


Figure 3.6. KITTI example, showing the ground truth 3D bounding box labels. [13]

3.2.2 TUM Traffic Intersection Dataset

TUM Traffic Intersection dataset (TUMTraf-I) focuses on urban traffic, especially at intersections, capturing the dynamics of complex urban scenes. Its use of both camera and LiDAR sensors allows for detailed perception studies. A notable weakness is its relatively smaller scale and lesser diversity in weather and lighting conditions compared to larger datasets, which may limit the generalizability of the findings derived from it [14].

Figure 3.7. TUM Traffic Interstate Examples from the two interstate angles, **Left:** showing the 3D bounding boxes and pointcloud groundtruth labels, **Right:** illustrating the ground truth 3D bounding boxes and tracking pattern of each labeled vehicle. [15]

3.2.3 Lyft

The Lyft dataset provides high-resolution 3D annotations across multiple cities, making it valuable for urban navigation tasks such as 3D object detection and prediction. While it offers detailed annotations and a multi-sensor setup, the dataset is less known for its temporal coverage, possibly affecting research in areas requiring time-series data, such as object tracking or behavior prediction [16].

Figure 3.8. Lyft L5 dataset Example from ego vehicle, **Left:** original image without labels **Right:** The image and point cloud data collected. [17]

3.2.4 nuScenes

Developed by Caesar et al. [6], the nuScenes dataset offers extensive multi-modal sensor data across various urban locations and conditions, supporting a wide range of tasks from detection to segmentation. Its comprehensive urban coverage is a significant strength. However, its LiDAR point clouds are sparser compared to newer datasets like Waymo, which might affect the performance of perception algorithms that rely on high-density point data.

Figure 3.9. nuScenes Example from ego vehicle, **Left:** Birds-eye-view LiDAR image with labeled ground truth 3D bounding boxes **Right:** View from the Back camera with the labeled ground truth 3D bounding boxes. [18]

3.2.5 Waymo Open Dataset

The Waymo Open Dataset is renowned for its large-scale and high-resolution sensor data. It addresses some limitations of previous datasets by including more diverse driving conditions and geographic areas. Nevertheless, while it provides a vast amount of data, the challenges of processing and extracting useful insights due to its sheer size and complexity can be a limiting factor for some research groups without substantial computational resources [19].

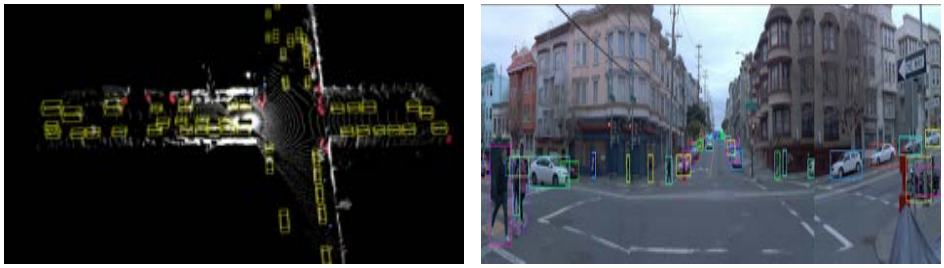


Figure 3.10. Waymo Example from ego vehicle, **Left:** Birds-eye-view LiDAR image with labeled ground truth 3D bounding boxes **Right:** View from front, front left and front right cameras with the labeled ground truth 2D bounding boxes. [20]

Dataset Name	Size	Scenes	3D Labels	Map	Annotations	Task
KITTY [13]	6h	50	80k	None	3D bounding boxes	Perception
TUMTraf-I [14]	0.305h		57.4k	None	3D bounding boxes	Perception
Lyft [16]	2.5h	366		Rasterised road geometry	3D bounding boxes	Perception
nuScenes [6]	6h	1000	1.4M	Rasterised road geometry	3D bounding boxes, trajectories	Perception, Prediction
Waymo [19]	10h	1000	12.6M	None	3D bounding boxes	Perception

Table 3.1. Details from datasets.

These datasets collectively advance the field of autonomous driving by providing diverse, high-quality data that enable the development of robust perception systems. They address different aspects of autonomous driving, such as urban navigation, intersection handling, and sensor integration, facilitating advancements in both academic research and practical applications. The continuous expansion of these datasets reflects the evolving challenges in autonomous driving, pushing forward the boundaries in machine learning and computer vision technologies, an overview of the datasets can be seen in Table 3.1.

3.3 Active Learning

Within various contexts, the key element of AL lies in the development of appropriate query formulations that are well-suited to the problem at hand. There are primarily two types of query formations: Query synthesis and sampling, where sampling is subdivided into stream-based sampling and pool-based sampling.

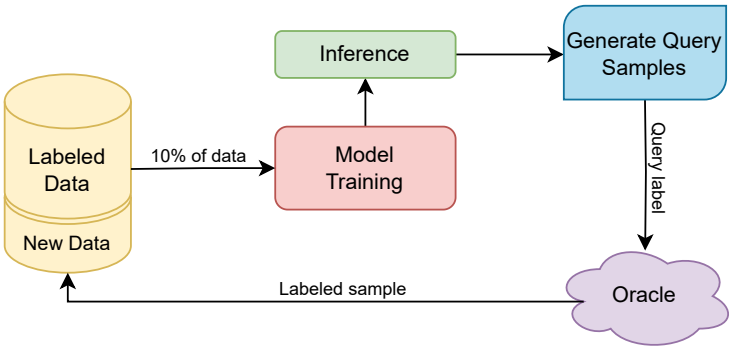


Figure 3.11. Membership Query Synthesis AL Scenario.

As depicted in Figure 3.11, the process known as membership query synthesis involves the model generating new synthetic data samples from its existing knowledge base. In this example initially, this knowledge base consists of the 10 % labeled data used for training. The model assesses areas within the data space where its predictions are uncertain, guiding the identification of where additional data would be most beneficial. Based on these uncertainties, the model—or a designated synthesizer—creates new data points aimed at resolving these ambiguities. This may involve modifying existing data points or crafting entirely new examples from the model's understanding of the feature space. While this method can efficiently generate numerous queries, its effectiveness is limited for complex tasks like natural language processing (NLP) or 3D detection, where synthetic samples may be challenging for human experts to interpret.

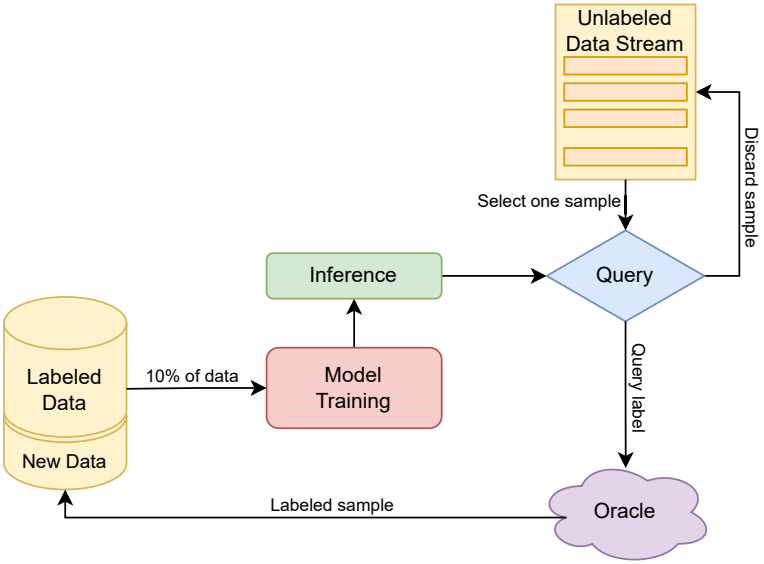


Figure 3.12. Stream-based sampling AL scenario.

As shown in Figure 3.12, the stream-based selective sampling involves presenting unlabeled data samples in a sequential, continuous flow. The model decides in real-time whether each sample should be labeled, employing an AL querying technique. Samples that meet the querying criteria are selected for further inquiry by the oracle, while others are discarded as the model moves on to assess subsequent samples.

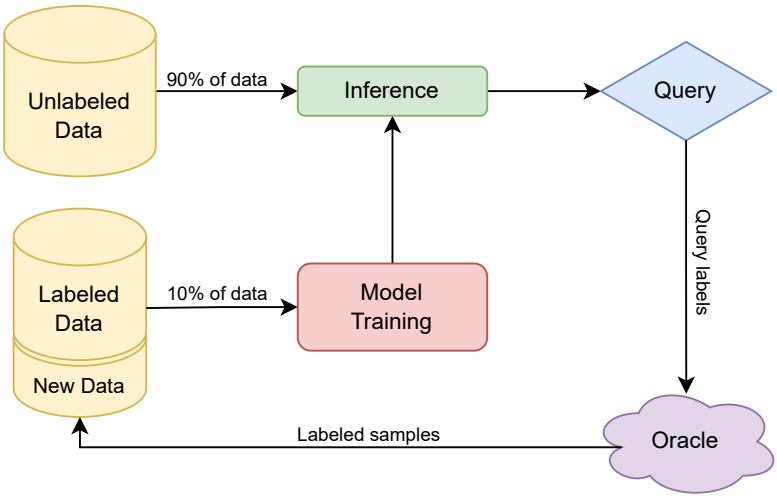


Figure 3.13. Pool-based sampling AL scenario.

A commonly used AL scenario, known as pool-based sampling and depicted in Figure 3.13, the model starts with a large pool of unlabeled data samples. The learner selects samples for inquiry based on a hypothesis model constructed from initially labeled samples. Each sample, once labeled by an oracle, augments the labeled data pool, prompting re-training of the model. This approach is well-suited for scenarios equipped with ample computational and storage capabilities, enabling an iterative process of training, querying, and model refinement. The process persists until labeling resources are fully utilized or a specific performance threshold is reached.

As pool-based sampling is the most used strategy, this section will continue elaborating on the specific approaches of the strategy. Based on the sample selection criteria, Pool-based AL can be further split into three categories, often referred to as the query strategy or the acquisition function.

The first category, known as uncertainty sampling, is a foundational approach in AL. In this method, the model selects instances where its predictive confidence is notably low, aiming to enhance the accuracy of its learning by focusing on the most ambiguous data points. Several techniques fall under this method. [9]

One such approach, Bayesian Active Learning by Disagreement (BALD) [21], implements Bayesian neural networks to perform approximate variational inference. The core idea behind BALD is to select data points that are likely to maximize the mutual information between the model's predictions and its parameters. Essentially, these are data points that, when learned, are expected to contribute the most to the informational content of the model parameters.

As a visting gratuate student at the LISA laboratory [22], I contributed to the development of new methodologies in the field. Specifically, I played a pivotal role in creating the ActiveAnno3D method [23], which applies uncertainty techniques to optimize the AL process in multi-modal 3D object detection. This method focuses on selecting highly informative samples for labeling using an entropy querying approach. The ActiveAnno3D framework also incorporates continuous training methods to effectively balance computational demands

with detection performance. Notably, this approach has demonstrated that it can achieve near-optimal performance using only half the typically required training data, substantially lowering both annotation effort and costs.

Additionally, I was part of an analytical study [24] aimed at addressing the challenge of unbalanced datasets. This research explored the use of entropy querying to significantly diminish the performance disparities between majority and minority classes. The findings highlight the superiority of entropy querying over random sampling, proving its efficacy in optimizing resource allocation during model training, particularly in scenarios with limited data availability. This study further underscores the effectiveness of this method in balancing class representation and enhancing detection accuracy in a variety of complex and demanding driving scenarios.

The second category, diversity sampling, adopts a comprehensive approach to query strategy by aiming to explore the data space as extensively as possible. This method selects instances that are diverse or uncommon within the existing training set, to broaden the model's comprehension of the entire data landscape [9].

CoreSet [25], is a diversity-based AL method which is redefined as a core-set selection problem. This approach aims to select a subset of data such that a model trained on this subset is competitive over the entire dataset. Their method addresses the limitations of traditional AL heuristics when applied to convolutional neural networks (CNNs) in batch settings, where correlations between samples can diminish the effectiveness of these heuristics. By redefining the problem around core-set selection and providing a geometric bound to guide sample selection.

Hybrid sampling combines elements from both uncertainty and diversity sampling strategies, aiming to balance the trade-off between exploiting the model's current knowledge (exploitation) and exploring new, informative data points (exploration). This approach selects instances that are both uncertain and diverse, potentially offering a more balanced improvement in model performance across different dimensions of the data space. This combined strategy can be particularly effective in complex learning scenarios where neither uncertainty nor diversity sampling alone would be sufficient to improve model performance comprehensively. By integrating these strategies, hybrid sampling leverages the strengths of both to enhance the learning process more robustly [9].

One such notable approach is the Batch Active Learning by Diverse Gradient Embeddings (BADGE), developed by Jordan T. Ash et al. [26]. This method effectively integrates both predictive uncertainty and sample diversity within its operational framework. BADGE employs the concept of hallucinated gradients, which estimate the potential impact of each data point's label on the model parameters. These calculated gradients are instrumental in forming a batch of query points that are both uncertain and diverse. Additionally, the implementation of the k-MEANS++ initialization for selecting these points helps to ensure that the algorithm not only avoids redundant selections but also enhances the model's performance without the need for manual adjustments of hyperparameters.

Hypothesis 4

Throughout the *Related Works* section, several critical issues have been identified within the existing methodologies. The most prominent of these include:

- Dataset creation cost.
- Class imbalance within the datasets, which can skew the training process and model performance.
- Issues of class underfitting and overfitting, which hinder the model's ability to generalize effectively.
- The inadequacy of single-sensor modalities to provide the necessary accuracy for this domain in many cases.

In response to these challenges, the focus of this thesis will be on the development and evaluation of AL querying methods tailored to the diversity-based sampling strategy. This approach aims to address and mitigate the identified issues, particularly the limitations posed by class imbalance and under/overfitting, while a multi-modal model will be used in order to alleviate the sensor modality constraints.

The primary hypothesis of this research is that:

> *Utilizing a diversity-based sampling method for data selection will yield higher model accuracy compared to a random sampling method. This improvement is expected because the diversity-based approach selects samples based on specific criteria that enhance the representational completeness of the training set, unlike random sampling which lacks such a targeted selection mechanism.*

Technical Analysis 5

In this section, a detailed examination will be conducted of the selected architecture and the dataset that will be utilized for this project. The specific features, capabilities, and potential limitations of each architectural choice will be delved into, exploring how they align with the project's objectives. Additionally, the dataset's suitability will be assessed, including its composition, diversity, and relevance to the scenarios anticipated in practical applications. This analysis aims to provide a thorough understanding of the foundational elements that will drive the success of the project.

5.1 Model

As discussed in the related works section, there is a growing trend towards employing multi-modal model architectures for 3D object detection, with various approaches to constructing these architectures. For this project, collaborative partners have expressed a particular interest in the BEVFusion model. This model has consistently demonstrated a robust architecture that performs effectively across multiple datasets and offers significant advantages over other models.

5.1.1 BEVFusion

At the core of BEVFusion's methodology is the adoption of bird's-eye view (BEV) as the unifying representation space. This choice is instrumental in preserving the full semantic density of camera features alongside the precise geometric structure of LiDAR data, a combination that has been difficult to achieve with previous methods.

The process begins with distinct neural network encoders that extract features from camera and LiDAR data separately. This initial step is crucial, as it leverages the unique characteristics of each sensor modality, ensuring that the most relevant and critical information is captured efficiently.

Unlike traditional methods that force one sensor's data into the frame of reference of another, BEVFusion maps both camera and LiDAR features into a shared BEV space. This is achieved for the images through an optimized view transformation mechanism, while the LiDAR data is flattened.

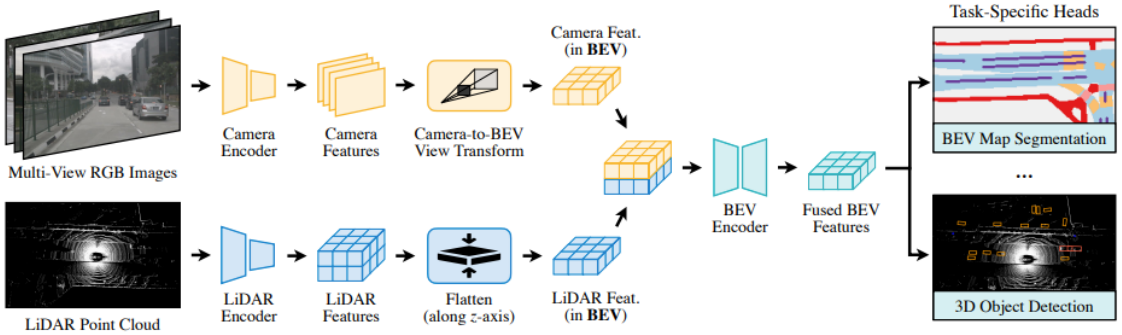


Figure 5.1. BEVFusion Architecture [4].

A critical bottleneck in prior sensor fusion approaches has been the computationally intensive process required to transform and pool features into a new representation. Addressing this challenge, BEVFusion introduces a specialized kernel for efficient BEV pooling of the images, which significantly accelerates this process without compromising accuracy.

The BEV pooling kernel employs two main strategies: precomputation and interval reduction. These strategies are designed to optimize the process of associating camera feature pixels with corresponding locations in the BEV space and aggregating these features efficiently.

The pre-computation step leverages the static nature of the camera's position relative to the vehicle, allowing the spatial relationship between camera pixels and their corresponding BEV grid locations to be calculated in advance. Since the camera intrinsics (focal length, principal point, etc.) and extrinsic (position and orientation relative to the vehicle body) remain constant, the mapping from the camera's perspective view to the BEV grid does not change dynamically. By pre-computing this mapping, the system can bypass the computationally intensive process of calculating the BEV grid location for each pixel during runtime. This results in a significant reduction in latency, as the pre-computed mappings can be quickly looked up and applied to the incoming camera data. [4]

Interval reduction optimizes the aggregation phase, where features within each BEV grid cell are combined (e.g., by averaging). Traditional approaches might compute aggregations such as sum or mean across all features in a grid cell by iterating through each feature or using prefix sum operations, which are inefficient at scale [27]. The interval reduction technique, however, assigns a dedicated GPU thread to each BEV grid cell, parallelizing the aggregation operation. Each thread computes the aggregate feature value for its respective cell independently, significantly reducing the computational overhead associated with tree reductions and memory access in conventional approaches.

Once the camera and LiDAR data have been transformed into BEV features, these features are integrated. Fusion is typically achieved through concatenation. However, this method may not fully address the occasional misalignment observed between LiDAR derived BEV features and Camera derived BEV features, attributed largely to the variable accuracy in the unsupervised depth estimation of the camera-BEV features. To rectify such discrepancies and ensure coherent alignment, a convolution-based BEV encoder is employed as a final step. This encoder effectively compensates for any local misalignment's between the feature sets. The architecture illustrating this process can be seen in Figure 5.1.

There are many benefits of adopting BEVFusion for sensor fusion. By maintaining the integrity of both geometric and semantic information, BEVFusion achieves superior performance in 3D object detection and BEV map segmentation. Additionally, the optimized BEV pooling and view transformation processes significantly reduce the computational demand. This efficiency does not come at the expense of accuracy or detail, making BEVFusion a highly effective solution for autonomous driving systems. Moreover, the versatility of the BEV representation space allows BEVFusion to adapt across different 3D perception challenges, offering a comprehensive solution that enhances the robustness of autonomous systems in a variety of environmental conditions, including low-light and adverse weather scenarios. [4]

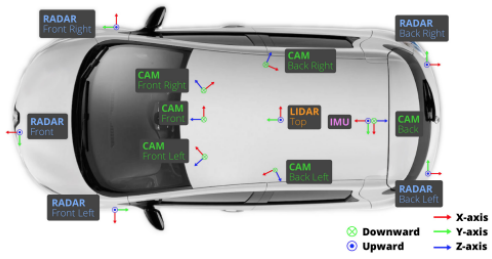
5.2 Dataset

When selecting a dataset, numerous factors must be considered. This project requires a dataset that contains real-world data, including extensive LiDAR and camera data. It is also crucial that the dataset features a wide variety of data, encompassing a diverse range of object classes, weather conditions, and other environmental variables. Given these requirements, the ideal choices would be either the nuScenes [6] dataset or the Waymo [19] dataset, as both meet these criteria. However, given that the Waymo dataset includes less LiDAR data compared to nuScenes, the nuScenes dataset has been selected for use in this project.

5.2.1 nuScenes

The nuScenes dataset, addresses the critical need for a comprehensive multi-modal dataset to advance autonomous driving technologies. nuScenes includes a full suite of autonomous vehicle sensors, offering a full view of the vehicle's surroundings through six cameras, five radars, and one LiDAR, with 360-degree coverage, as seen in Figure 5.2. This diverse sensor setup caters to the complexity of autonomous driving tasks by providing rich data for training and evaluating machine learning models on detection, tracking, and segmentation tasks under varied environmental conditions.

nuScenes comprises of data collected from Boston and Singapore, chosen for their challenging driving environments. The dataset includes 1,000 scenes, each 20 seconds long, annotated with 3D bounding boxes for 23 classes across more than 1.4 million images, 400k LiDAR sweeps, and 1.3 million radar sweeps. The LiDAR provides dense point clouds with accurate 3D localization, while the cameras capture detailed semantic information, and the radars offer long-range detection capabilities with velocity measurements. This rich sensor fusion enables a nuanced understanding and perception of dynamic environments essential for autonomous vehicles.



(a)

Sensor	Details
6x Camera	RGB, 12Hz capture frequency, 1/1.8" CMOS sensor, 1600 × 900 resolution, auto exposure, JPEG compressed
1x Lidar	Spinning, 32 beams, 20Hz capture frequency, 360° horizontal FOV, −30° to 10° vertical FOV, ≤ 70m range, ±2cm accuracy, up to 1.4M points per second.
5x Radar	≤ 250m range, 77GHz, FMCW, 13Hz capture frequency, ±0.1km/h vel. accuracy
GPS & IMU	GPS, IMU, AHRS. 0.2° heading, 0.1° roll/pitch, 20mm RTK positioning, 1000Hz update rate

(b)

Figure 5.2. Sensor setup for NuScenes data collection platform [6].

The annotation process in nuScenes is meticulously designed to ensure high-quality, reliable data for training and validating autonomous driving systems. Each of the 1,000 scenes in the dataset is fully annotated with 3D bounding boxes for 23 different classes, including a variety of vehicles, pedestrians, and environmental objects, providing a detailed understanding of the scene dynamics. Annotations also include eight attributes per object, such as visibility, pose, and activity, offering additional context that is crucial for nuanced perception and decision-making tasks in autonomous driving.

Annotators used a combination of manual and semi-automated methods to label the data, ensuring both accuracy and consistency across the dataset. The extensive annotation process results in a dataset that is not only large in volume but also rich in information, with 7 times as many annotations and 100 times as many images as the pioneering KITTI dataset. This amount of annotated data makes nuScenes a valuable resource for developing advanced detection and tracking algorithms.

Figure 5.3. Front camera images collected from clear weather (col 1), nighttime (col 2), rain (col 3) and construction zones (col 4) [6].

The nuScenes dataset is structured to facilitate easy access and manipulation of its rich multi-modal data. Annotations and metadata are stored in a relational database, avoiding redundancy and enabling efficient querying. The dataset includes detailed calibration data for all sensors, timestamps for synchronization, and precise vehicle localization, allowing researchers to reconstruct scenes accurately and analyze the data effectively.

Access to the dataset is further simplified by the nuScenes development kit (devkit), which provides tools for loading, visualizing, and evaluating data. The devkit includes APIs for Python, simplifying the integration of nuScenes into existing research workflows. The dataset is made available under a CC BY-NC-SA 4.0 license, encouraging widespread use in

the non-commercial research community. This approach to data accessibility ensures that nuScenes can be a valuable tool for a broad spectrum of research in autonomous driving. [6]

Since its release, nuScenes has facilitated a range of studies, contributing to advancements in 3D object detection and tracking under various weather conditions [5, 4]. The dataset has also spurred innovation in radar and sensor fusion research, areas that are critical for the reliability and robustness of autonomous systems [28, 29]. By providing a standardized benchmark with novel metrics for evaluation, nuScenes has not only pushed the state-of-the-art forward but also helped unify the research community's efforts toward solving common challenges in autonomous driving [6].

5.2.2 nuScenes Metrics

The nuScenes dataset introduces a comprehensive metric, the nuScenes Detection Score (NDS), designed to holistically evaluate the performance of autonomous driving systems in object detection tasks. This metric reflects the complex nature of real-world autonomous driving scenarios by incorporating various factors essential for accurate and reliable object detection.

The NDS is a scalar score that combines the mean Average Precision (mAP) with five metrics, seen in Table 5.1.

mAP	Mean Average Precision
$mATE$	Mean Average Translation Error
$mASE$	Mean Average Scale Error
$mAOE$	Mean Average Orientation Error
$mAVE$	Mean Average Velocity Error
$mAAE$	Mean Average Attribute Error

Table 5.1. NuScenes metrics used to create the NDS score.

Each metric addresses a specific aspect of detection quality, such as the precision in object localization (ATE), the accuracy in estimating object size and shape (ASE), the correctness in determining object orientation (AOE), the accuracy in predicting object velocity (AVE), and the precision in recognizing object attributes (AAE).

NDS is calculated as a weighted sum, where half of the score is based on mAP, reflecting the detector's ability to correctly identify and localize objects across different classes and scenarios. The other half is derived from the inverse of the mean errors of the metrics, emphasizing the importance of not just detecting objects but also accurately characterizing their state and dynamics. This dual focus ensures that the NDS provides a balanced measure of a detection system's overall performance, capturing both its detection capabilities and its precision in object characterization.

By integrating multiple dimensions of detection performance into a single metric, the NDS offers several advantages for benchmarking in autonomous driving research. It encourages the development of detection systems that are not only accurate in identifying objects but also precise in capturing their attributes and states, which are crucial for safe and effective

autonomous navigation. Furthermore, the NDS facilitates direct comparisons between different detection approaches, promoting transparency and progress in the field. [6]

5.3 3D Annotation

In this section, an in-depth explanation will be given of concepts like 3D labeling of objects, based on the ASAM OpenLABEL [30] standard and the nuScenes Labeling Scheme [31].

5.3.1 Coordinate Systems

A data stream formatted in OpenLABEL can carry sensor data from diverse sources alongside a wide range of associated labeling information. It is important to clearly define the connections across a varied dataset, these could be the interplay between different sensor outputs, the linkage between label data and sensor data, and how sensor data corresponds to real-world scenarios. This clarity is achieved through the application of multiple coordinate systems and transitions between them, ensuring the integrity of these relationships. For example, such a data stream might feature imagery from six cameras and point cloud outputs from one LiDAR sensor, as is the case with the nuScenes dataset [6]. The sensors can either be egocentric (mounted on a vehicle) or integrated into traffic infrastructure [6, 14, 19].

For this, the key concepts are:

- **Coordinate system:** Utilized to accurately specify point locations within a space, ranging from the two-dimensional positioning of a pixel in an image to the three-dimensional placement of a LiDAR point relative to a designated global origin. This origin might be defined as the vehicle's center-of-gravity (CoG), the rear axle, or the position of the LiDAR sensor itself. Commonly, these frameworks adopt the form of 3D right-handed Cartesian systems, as illustrated in Figure 5.4.

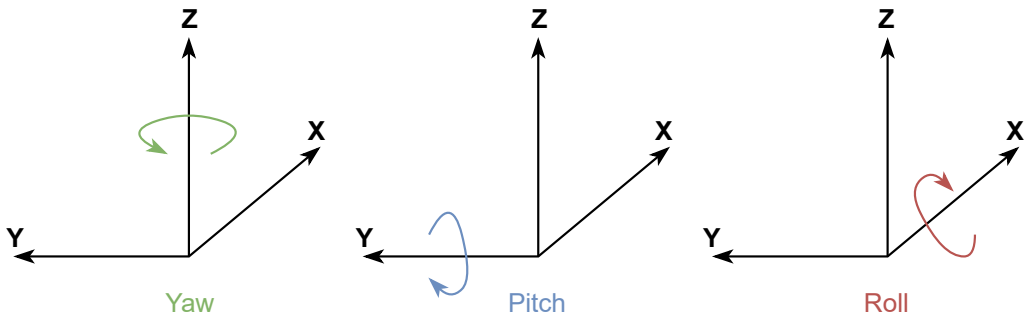


Figure 5.4. 3D right-handed Cartesian System.

- **Transformation**: These are crucial mathematical operations that enable the translation of points between different coordinate systems, ensuring the spatial integrity of these points remains intact across various representations. This capability is fundamental in accurately representing the same physical location in multiple spatial models. Primarily, two types of transformations are of importance:

1. **Camera Transform**: A projective transformation that facilitates the mapping of a three-dimensional point in the real world onto a two-dimensional pixel on a camera sensor. In order to succeed in this process it is vital to consider several factors, these being:
   - **Intrinsics**: These are parameters intrinsic to the camera that influence how light is projected onto the sensor. Key intrinsic factors include the focal length, which determines the zoom level and perspective effect, and the optical center, indicating the sensor point directly in line with the lens' line of sight.
   - **Distortion Coefficients**: Necessary to correct lens distortions that warp images, particularly at the edges, due to imperfections in the lens shape or material.
   - **Extrinsics**: Describe the camera's physical orientation and position in space, enabling the accurate depiction of objects on the sensor based on their real-world locations.
2. **Cartesian Transform**: This transformation is applied to convert coordinates from a 3D Cartesian system—used commonly for local spatial measurements—to an ellipsoidal coordinate system as utilized by Global Navigation Satellite Systems (GNSS). The 3D Cartesian system describes points in X, Y, and Z dimensions from a reference point, simplifying calculations. In contrast, the ellipsoidal system, accounting for the Earth's curvature, specifies locations in terms of latitude, longitude, and sometimes elevation. This conversion is essential for accurately representing positions on the Earth's surface, especially over distances where the planet's curvature significantly impacts measurements.

These transformations play a pivotal role in a variety of applications, such as sensor fusion, accurately creating maps from satellite imagery, and enabling autonomous vehicles to navigate by understanding their environment. By facilitating precise conversions and comparisons of spatial data, they ensure coherence across measurements from different sources or formats, crucial for tasks requiring high levels of spatial accuracy. [30]

5.3.2 General Geometry of Labels

Assigning labels to objects in datasets involves adopting geometric strategies that align with the nature of the sensor in question. Depending on whether the sensor data is two-dimensional or three-dimensional, appropriate geometric methods must be applied. The OpenLABEL Standard provides an array of geometric primitives designed to facilitate the labeling of objects and areas within these sensor streams. In this context, this analysis is centered on the application of both 2D and 3D bounding boxes for labeling purposes.

In Figure 5.5 a 2D bounding box can be seen. It is defined as a rectangle, with an array of four or five floating points, this definition depends on whether a rotation is specified or not.

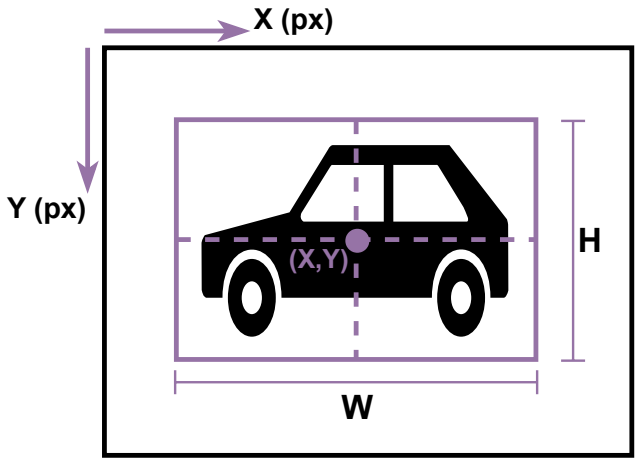


Figure 5.5. 2D Bounding Box definition.

X	x-coordinate of the center of the rectangle.	[Pixel]
Y	y-coordinate of the center of the rectangle.	[Pixel]
W	Width of the rectangle in the x/y-coordinate system.	[Pixel]
H	Height of the rectangle in the x/y-coordinate system.	[Pixel]
Alpha	Rotation of bounding box as a right-handed rotation, implies a positive rotation from x-axis to y-axis. The point of origin of the rotation point is the center of the bounding box (optional).	[Radians]

A 3D bounding box is defined as a cuboid within a three-dimensional Euclidean space, distinguished by its position, rotation, and dimensions. The box's position and dimensions are denoted by three-dimensional vectors as seen in Figure 5.6. Rotation can be represented in one of two ways, either through a four-vector quaternion or by using a three-vector Euler approach. When using Euler angles, the sequence of ZYX is preferred, aligning with the order of yaw-pitch-roll. [30]

Position	x, y and z coordinates of the 3D position of the center of the cuboid.	[meters]
Rotation	(Four-vectors) Quarterion in non-unit form (x, y, z and w).	[.]
Rotation	(three-vectors) Euler angles yaw, pitch and roll as rz, ry and rx respectively.	[.]
Dimension	x, y and z dimensions of the cuboid or the x-coordinate.	[meters]

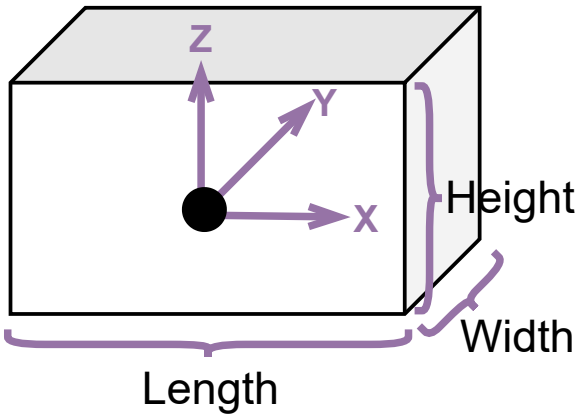


Figure 5.6. 3D Bounding Box definition.

5.3.3 nuScenes Annotation Scheme

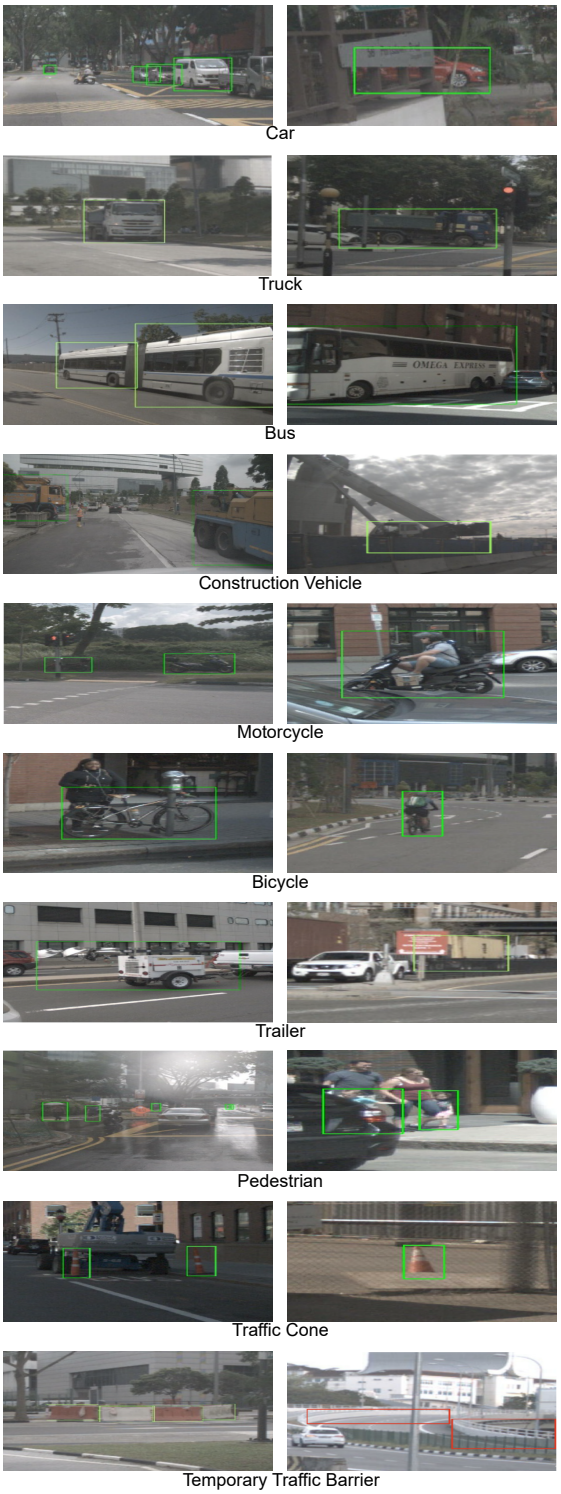
As mentioned earlier in section 5.2.1, the nuScenes dataset contains 1.4 million annotated objects across 23 distinct classes. Of these, 10 classes are specifically utilized within the BEVFusion model [4]. This section will delve into the annotation guidelines for these 10 classes. For details on annotating the remaining classes, readers are directed to [31].

The dataset employs cuboid 3D bounding boxes for annotations, illustrated in Figure 5.3. Given that the annotations occur over data from three different sensors(Camera, LiDAR, Radar), achieving a high level of precision in drawing these boxes is crucial. This ensures that no information is overlooked or excluded from other data formats.

However, to clearly demonstrate the categorization of each object type, green 2D bounding boxes will be utilized to identify example objects that fit within each class. While, red 2D bounding boxes will indicate objects that should not be annotated as part of a specific class.

The criteria for class annotations are outlined as follows:

- **Car/Van/SUV**: Vehicle designed primarily for personal use, e.g. sedans, hatch-backs, wagons, vans, mini-vans, SUVs and jeeps.
- **Truck**: Vehicles primarily designed to haul cargo including pick-ups, lorrys, trucks and semi-tractors. Trailers hauled after a semi-tractor should be labeled as "Trailer".
  - **Pickup Truck**: A pickup truck is a light duty truck with an enclosed cab and an open or closed cargo area. A pickup truck can be intended primarily for hauling cargo or for personal use.
  - **Front or Semi Truck**: Tractor part of a semi trailer truck. Trailers hauled after a semi-tractor should be labeled as a trailer.
- **Bus**:
  - **Bendy Bus**: Buses and shuttles designed to carry more than 10 people and comprises two or more rigid sections linked by a pivoting joint. Annotate each section of the bendy bus individually.
  - **Rigid Bus**: Rigid buses and shuttles designed to carry more than 10 people.
- **Construction Vehicle**: Vehicles primarily designed for construction. Typically very slow moving or stationary. Cranes and extremities of construction vehicles are only included in annotations if they interfere with traffic. Trucks used to hauling rocks or building materials are considered trucks rather than construction vehicles. [31]



- **Motorcycle**: Gasoline or electric powered 2-wheeled vehicle designed to move rapidly (at the speed of standard cars) on the road surface. This category includes all motorcycles, vespas and scooters. It also includes light 3-wheel vehicles, often with a light plastic roof and open on the sides, that tend to be common in Asia. If there is a rider and/or passenger, include them in the box.
- **Bicycle**: Human or electric powered 2-wheeled vehicle designed to travel at lower

speeds either on road surface, sidewalks or bicycle paths. If there is a rider and/or passenger, include them in the box.

- **Trailer**: Any vehicle trailer, both for trucks, cars and motorcycles (regardless of whether currently being towed or not). For semi-trailers (containers) label the truck itself as "Truck".
- **Pedestrian**:
  - **Adult**: An adult pedestrian moving around the cityscape. Mannequins should also be annotated as Adult Pedestrian.
  - **Child**: A child pedestrian moving around the cityscape.
- **Traffic Cone**: All types of traffic cones.
- **Temporary Traffic Barrier**: Any metal, concrete or water barrier temporarily placed in the scene in order to re-direct vehicle or pedestrian traffic. In particular, includes barriers used at construction zones. If there are multiple barriers either connected or just placed next to each other, they should be annotated separately. [31]

Although the nuScenes dataset represents a significant advancement with an entire sensor suite implementation and extensive annotations of various data types, it faces the common issue of class imbalance that plagues many autonomous vehicle datasets. This imbalance arises because certain objects, such as cars and pedestrians, are more frequently encountered on the roads compared to others like bicycles(in Boston and Singapore) and construction vehicles. The skewed distribution of these classes is evident in Table 5.2.

Category	NuScenes Cuboids	Cuboid Ratio	LiDARseg Points	Point Ratio
Car	493,322	42.30%	38,104,219	48.27%
Truck	88,519	7.59%	15,841,384	20.07%
Bus	16,321	1.4%	4,604,760	5.83%
Construction Vehicle	14,671	1.26%	1,514,414	1.92%
Motorcycle	12,617	1.08%	427,391	0.54%
Bicycle	11,859	1.02%	141,351	0.18%
Trailer	24,860	2.13%	4,907,511	6.22%
Pedestrian	222,164	19.05%	2,344,427	2.73%
Traffic Cone	97,959	8.40%	736,239	0.93%
Temporary Traffic Barrier	152,087	13.04%	9,305,106	11.79%
Full Dataset Total	1,166,187	100.00%	78,942,623	100.00%
BEVFusion Classes Total	**1,134,379**	**97.27%**	**77,926,802**	**98.48%**

Table 5.2. The distribution of the 10 classes used by BEVFusion compared to a total of 23 classes. [12]

Table 5.2 clearly illustrates the disparity in annotation volume across different classes. This imbalance presents two major issues. The first issue is the potential difficulty in accurately identifying objects from sparsely annotated classes. The second issue is the increased likelihood of the model overfitting to classes that are heavily annotated.

5.4 Algorithm Architectures

There are many compelling implementations of AL diversity-based methods, as explored in Section 3.3. While these existing methods demonstrate significant benefits and hold considerable promise, there is also merit in investigating new and unconventional

technologies. Exploring these avenues can further advance understanding of these methods and potentially direct new research toward innovative directions.

As part of this incentive, this thesis will focus on creating a diversity-based AL method which utilizes hierarchical clustering and vision-language models (VLMs) to effectively query new samples.

VLMs have become prominent, after the success of Language Models (LMs) which are the foundation of modern natural language processing (NLP). These models are designed to understand, generate, and interpret human language based on the statistical properties of text data. By leveraging vast amounts of text, language models learn the intricacies of language—from syntax and semantics to context and colloquialisms. Traditional models, such as n-gram models [32, 33], have evolved into more sophisticated neural network architectures like recurrent neural networks (RNNs) [34] and, more recently, transformers [35]. These advanced models, exemplified by GPT (Generative Pre-trained Transformer) [36, 37], excel in tasks ranging from text completion to complex question answering, demonstrating a nuanced grasp of language patterns and usage. The general architecture of one such model can be viewed in Figure 5.7.

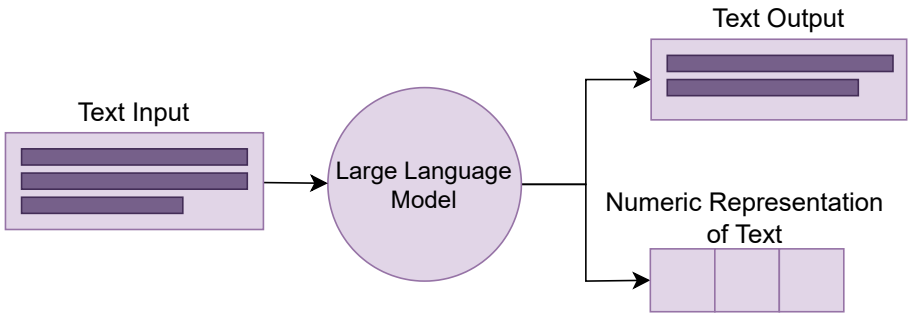


Figure 5.7. Overview of a Large Language Model.

VLMs are a type of model where visual data and language are jointly interpreted. These models integrate the modalities of vision and language, aiming to understand and generate descriptions of visual content, answer questions about images, and even engage in dialogue about visual scenes, one such architecture is seen in Figure 5.8. This integration allows the models to perform tasks such as image captioning, visual question answering, and cross-modal retrieval. A prominent example is the CLIP (Contrastive Language-Image Pre-Training) model, which learns visual concepts from natural language descriptions, enabling it to understand images in a way that mirrors human visual and linguistic capabilities. By training on a diverse range of images paired with textual descriptions, VLMs like CLIP develop a robust understanding that supports both high-level reasoning and detailed image analysis [38].

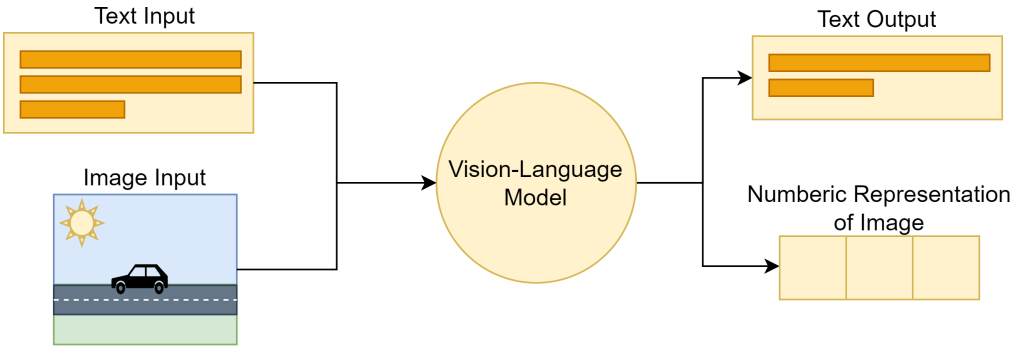


Figure 5.8. Overview of a Vision-Language Model.

Therefore, the Contrastive Language-Image Pre-Training (CLIP) Model is selected as the VLM architecture for this problem. The model is chosen because of its proven capability to classify objects accurately without the necessity for additional training.

5.4.1 Contrastive Language-Image Pre-Training (CLIP) Model

Traditional computer vision systems, despite their effectiveness, are often limited by their reliance on fixed, predetermined object categories. This limitation necessitates additional labeled data for new visual concepts, thereby constraining their generalizability and applicability. CLIP emerges as a promising solution to these challenges by learning visual concepts directly from natural language descriptions, leveraging the vast amount of text available on the internet.

The evolution of computer vision systems has predominantly been driven by models trained to recognize a fixed set of predetermined object categories. This approach, while effective for specific tasks, inherently limits the system's applicability and adaptability to new or unforeseen visual concepts. Traditionally, extending the capabilities of these systems to new categories or tasks requires gathering and labeling a new dataset, a process that is both time-consuming and resource-intensive.

The advent of natural language processing (NLP) technologies, particularly advanced pre-training methods, has unveiled the potential of using the vast corpus of text available on the internet as a rich source of supervision. This shift in perspective suggests an alternative paradigm for computer vision: learning directly from raw text descriptions of images. This method not only circumvents the need for task-specific dataset creation, but also enables models to learn a broader range of visual concepts in a more flexible and scalable manner.

CLIP represents a novel approach to bridging the gap between visual understanding and natural language processing. At its core, CLIP is designed to learn visual concepts from natural language descriptions, allowing it to generalize across a wide range of visual tasks without task-specific training. The model architecture is built upon two main components: a visual model that processes images and a language model that interprets text descriptions. These components are trained simultaneously using a contrastive learning objective that aligns the image and text embeddings in a shared multidimensional space.

The primary innovation behind CLIP lies in its pre-training task: predicting the most probable pairing of a batch of images and text descriptions from a dataset. This task

leverages a dataset of 400 million (image, text) pairs, enabling CLIP to learn a vast array of visual concepts directly from natural language supervision. The result is a model that can perform zero-shot transfer to a variety of downstream tasks, demonstrating state-of-the-art image representations learned entirely from scratch.

To enable the contrastive pre-training approach, there are three key elements, these are:

Data Collection: To support the approach, the authors behind the CLIP model create the WebImageText (WIT) dataset, a collection of 400 million (image, text) pairs curated from a diverse amount of publicly available sources on the internet. This is done to cover as broad a set of visual concepts as possible. To gather data points, the authors use 500.000 predetermined queries, where each query will contain up to 20.000 (image, text) pairs. This vast and diverse dataset is the foundation for CLIP's flexible and scalable learning capabilities.

Model Training: In the pre-training phase, the primary objective is to predict whether a given image and text pair are associated. This is accomplished by jointly training an image encoder and a text encoder using a contrastive learning approach, as seen in Figure 5.9. The large-scale dataset is crucial in this as it provides diverse and extensive supervision. During contrastive learning, the image encoder processes the image to produce a feature embedding, and the text encoder processes the text to produce a corresponding feature embedding. The core idea is to maximize the cosine similarity between the feature embeddings of correct (image, text) pairs and minimize it for incorrect pairs. The similarity score is scaled by a temperature parameter and normalized using a softmax function.

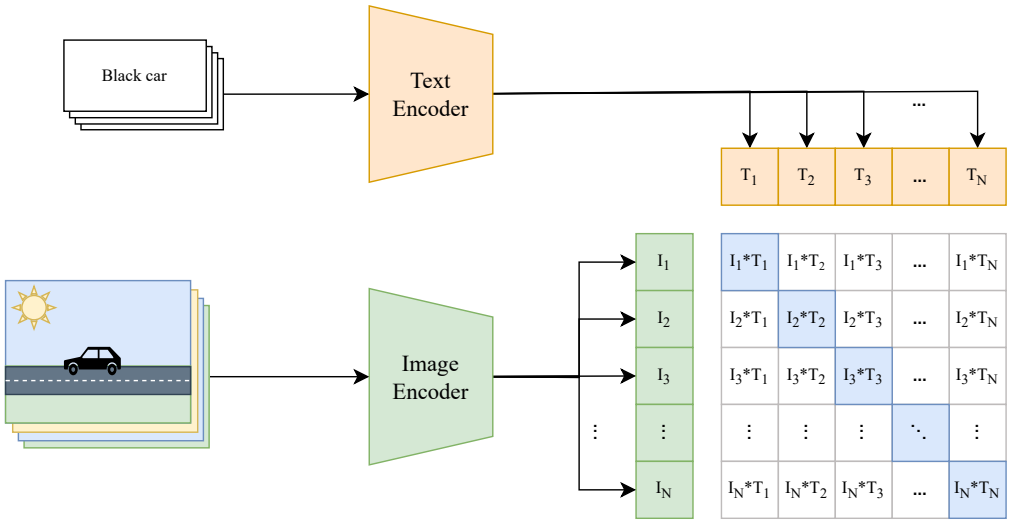


Figure 5.9. Contrastive Pre-training.

This approach allows CLIP to efficiently learn from natural language supervision, scaling effectively with the size of the pre-training dataset. The simplicity of this objective, combined with the model's ability to learn from a large and diverse dataset, underpins CLIP's exceptional performance across a range of visual tasks.

Evaluation (Zero-Shot Learning): After pre-training, the model can be used for zero-shot classification. For a given dataset, the names of all classes are used as text inputs, and

each class name is encoded into a feature embedding using the pre-trained text encoder. An image from the dataset is processed by the pre-trained image encoder to generate its feature embedding. The cosine similarity between the image embedding and each class name embedding is computed, this process can be viewed in Figure 5.10. These similarity scores are then scaled and transformed into probabilities using softmax. The class with the highest probability is selected as the predicted label for the image. [38]

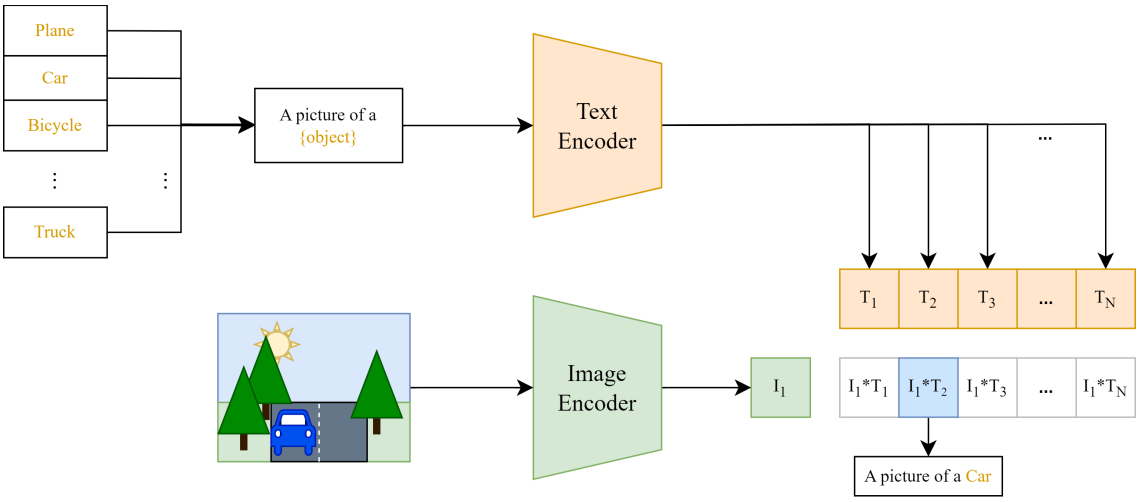


Figure 5.10. Zero-Shot learning on learned text embeddings and an unknown image.

5.4.2 Hierarchical Clustering

In order to calculate how diverse each sample is, a clustering method will be used called hierarchical clustering. It is a widely used unsupervised technique, which groups objects into clusters based on similarity. Within this method, each cluster is composed of data that shares characteristics, distinctly separating it from other established clusters.

A significant advantage of hierarchical clustering is its flexibility, it does not require the number of clusters to be specified in advance. Additionally, the method organizes the samples from the dataset into dendrograms, which simplifies the analysis and examination of the resulting clusters.

Hierarchical clustering can be implemented via two primary strategies:

- **Agglomerative**: This bottom-up approach begins with each sample as an individual cluster and merges them step-wise until one comprehensive cluster remains.
- **Divisive**: In contrast, this top-down approach starts with a single cluster that encompasses all samples, which is then divided step-by-step until every sample is isolated into its own cluster.

Figure 5.11 illustrates that the two clustering methods are essentially similar, differing only in their starting points. Given this similarity, this thesis will concentrate on just one approach. The Agglomerative method, being the most commonly used, will be the primary hierarchical clustering technique employed.

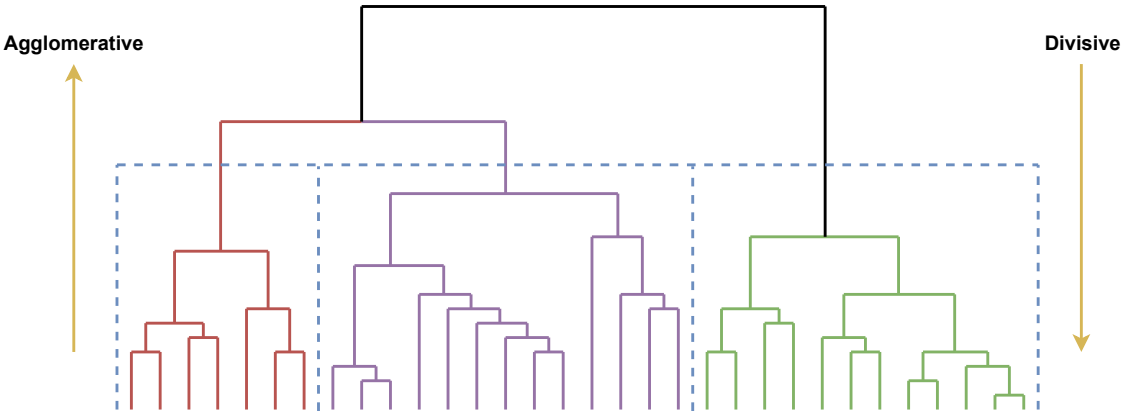
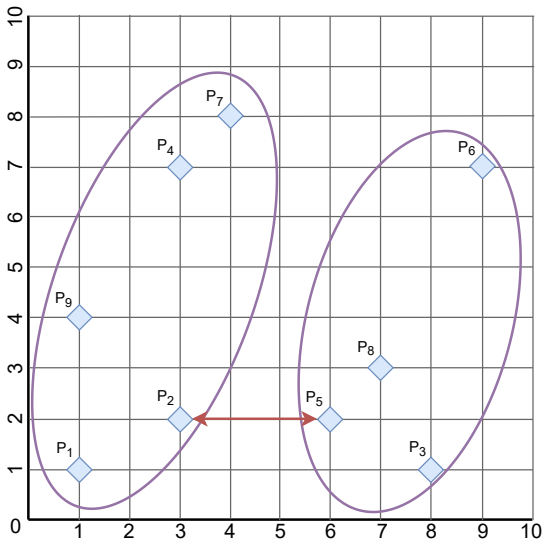


Figure 5.11. Hierarchical clustering diagram showing the bottom-up (Agglomerative) approach and top-down (Divisive) approach. The gray dashed line indicates a possible threshold, where the data would be split into three clusters in this case.

To successfully implement Agglomerative clustering on a dataset, certain procedural steps are required. Initially, a distance matrix must be constructed for all the clusters. Following this, a linkage operation is applied to the clusters. This operation relies on specific criteria that are tailored to address the problem at hand. Using two clusters as an example, the criteria include:

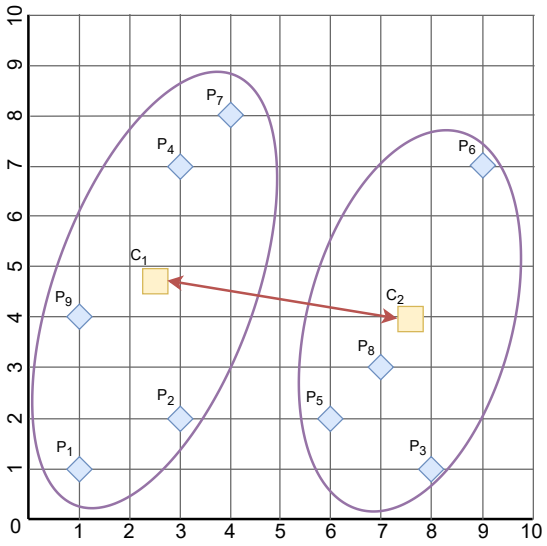
- **Single Linkage**: Determines the distance as the shortest distance between any two points within these clusters, seen in Figure 5.12.



Point	X	Y
P_1	1	1
P_2	3	2
P_3	8	1
P_4	3	7
P_5	6	2
P_6	9	7
P_7	4	7
P_8	7	3
P_9	1	4

Figure 5.12. Single Linkage.

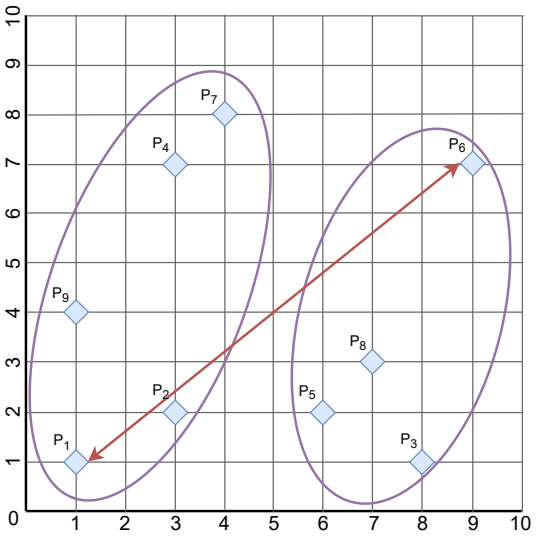
- **Average Linkage**: The distance is calculated as the average distance between all points in the first cluster and all points in the second cluster, shown in Figure 5.13.



Point	X	Y
P_1	1	1
P_2	3	2
P_3	8	1
P_4	3	7
P_5	6	2
P_6	9	7
P_7	4	7
P_8	7	3
P_9	1	4

Figure 5.13. Average Linkage.

- **Complete Linkage**: Calculates the distance based on the maximum distance between any two points in the clusters, illustrated in Figure 5.14.



Point	X	Y
P_1	1	1
P_2	3	2
P_3	8	1
P_4	3	7
P_5	6	2
P_6	9	7
P_7	4	7
P_8	7	3
P_9	1	4

Figure 5.14. Complete Linkage.

These linkage techniques are implemented based on a specific threshold set for the problem. A lower threshold ensures higher similarity among the samples within each cluster, but setting it too low may result in false negatives. Conversely, a higher threshold might simplify cluster formation but increase the risk of false positives. Thus, it is crucial to establish an optimal threshold to balance these outcomes effectively. [39, 40]

Algorithm Development 6

When examining the selected architecture, as discussed in Chapter 4, it becomes evident that the hypothesis, which is derived from the related works outlined in Chapter 3, applies to several of the most effective and recent methods analyzed in Chapter 5.

Therefore, this thesis presents a new diversity-based AL algorithm that utilizes clustering techniques alongside language models to enhance the learning process. By selecting a diverse array of samples, this approach aims to balance the disparities in class sizes. This strategy seeks to improve accuracy among less-represented classes without oversampling the more prevalent ones, hence addressing the potential issues of overfitting and underfitting.

6.1 Input: Dataset Pre-Processing

Numerous pre-processing techniques can be applied to images to prepare them for further analysis. Among these methods, recent research highlights the effectiveness of vision-language representations, particularly feature embeddings, in identifying unusual patterns and novelties within datasets [41]. Motivated by these findings, this thesis adopts such feature embeddings as the central pre-processing strategy for the nuScenes dataset.

To generate these feature embeddings, the CLIP model will be utilized. This model produces feature embeddings by converting images into dense matrices that capture a wide array of visual features. These matrices effectively encode the visual information in a format that is compatible with linguistic data, allowing for a multi-modal approach to understanding the content of the images. Each matrix represents the features of an individual image sample and is designed to facilitate the detection of nuanced patterns and anomalies.

These matrices will serve as the inputs for the novel algorithm developed in this thesis, ensuring that it benefits from a rich, feature-dense representation of each image. This approach enhances the algorithm's ability to make accurate identifications based on visual data.

6.2 Active VisLED-Querying Algorithm

The nuScenes dataset encompasses a diverse array of scenarios. Hence, it is crucial to approach the patterns identified by annotators with an open mind. A machine learning model, with its unique analytical capabilities, may uncover new patterns that are not immediately apparent to human observers. Such insights can provide a deeper understanding of the data, highlighting the potential of advanced analytics in revealing complex, unseen relationships within the input embeddings.

Therefore, the Active Vision-Language Enbedded Diversity Querying (VisLED-Querying) Algorithm is created with two scenarios in mind, the Open-World Exploring Scenario and the Closed-World Mining Scenario.

6.2.1 Open-World Exploring

The Open-World Exploring VisLED-Querying Scenario employs a novel approach by conducting hierarchical clustering directly on feature embeddings, as illustrated in Figure 6.1, bypassing the conventional method of categorizing them into predefined dataset classes. This technique enables a comprehensive analysis of the entire dataset, facilitating the identification of both broad patterns in the data and diverse samples, that transcend the limitations of existing class definitions.

Algorithm 1: Open-World Exploring VisLED-Querying

Input: Unlabeled pool of egocentric driving scene images

Output: Updated training set

1 Embed each egocentric driving scene image from the unlabeled pool using CLIP;

2 Use hierarchical clustering to separate the embeddings;

3 Sample new data points from the unclustered set for addition to the training set;

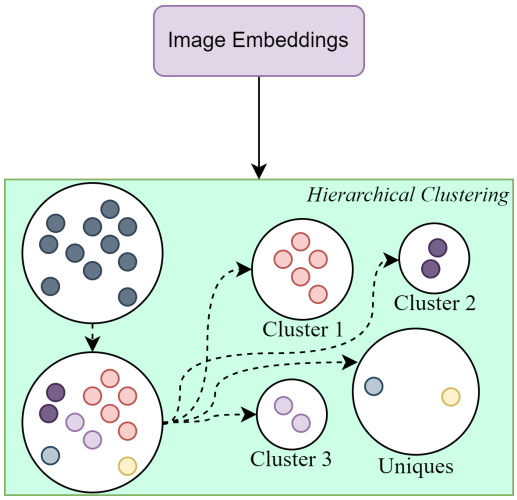


Figure 6.1. A representation of the Open-World Exploring algorithm, showing its pseudo code and architecture.

By avoiding the segmentation of data into predetermined classes, this method allows for a more comprehensive understanding of the dataset. It provides an opportunity to discover relationships and structures that might otherwise remain hidden within the confines of the predetermined classes. This approach is particularly advantageous in complex datasets like nuScenes, where diverse scenarios and interactions can manifest in ways that are not neatly aligned with established class categories.

6.2.2 Closed-World Mining

The Closed-World Mining VisLED-Querying algorithm adopts a similar framework to that of its counterpart, the Open-World Exploring VisLED-Querying algorithm, but introduces a critical modification: the initial segmentation of the dataset into predefined classes. This classification is achieved using the CLIP model, mirroring the approach taken in the open-world scenario but tailored to a more structured analysis environment.

In this closed-world setting, the algorithm employs zero-shot learning, a technique that enables the classification of data that the CLIP model has not explicitly encountered during training. Specifically, the algorithm will categorize each image sample based on the

most prominent object identified within the image. This categorization process involves comparing the visual features of the image with the textual descriptions of various classes, ultimately assigning the image to the class that exhibits the highest similarity score.

Once the initial classification is complete, an equal number of samples will be selected from each class. The method can be seen in Figure 6.2 where it is possible to view the pseudo code and structure of the algorithm.

Algorithm 2: Closed-World Mining VisLED-Querying

Input: Unlabeled pool of egocentric driving scene images

Output: Updated training set

1 Embed each egocentric driving scene image from the unlabeled pool using CLIP;
2 Encode each class label using a text encoding;
3 Applying zero-shot learning by maximizing the product of the embeddings, sort the embedded images by class;
4 For each class, apply hierarchical clustering;
5 Sample new data points from the unclustered set associated with the desired class, and add to the training set;

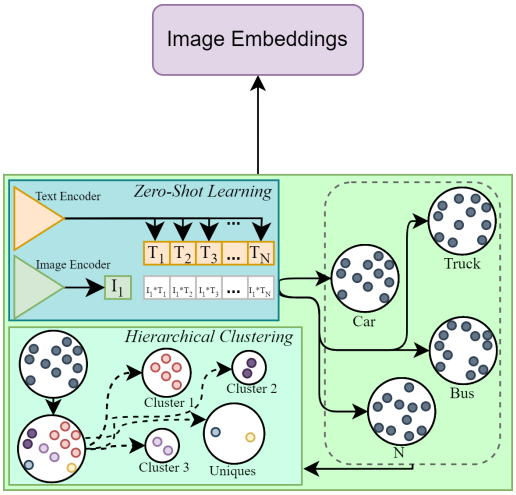


Figure 6.2. A representation of the Closed-World Mining algorithm, showing its pseudo code and architecture.

6.3 Output

Although the methods employed by the two algorithms differ, they ultimately generate identical types of output. The algorithm is executed multiple times, each corresponding to a different training split, corresponding to the chosen criteria. This process is designed to create distinct training subsets for each designated data split, thereby enhancing the robustness and diversity of the training regime.

In the course of operation, the algorithms focus on identifying and selecting the most diverse samples. This selection is achieved by prioritizing samples that exhibit unique characteristics which do not align closely with any existing cluster. Once the diverse sample has been chosen, the scene name of the sample will be saved, if a scene name has already been included in a chosen subset, that particular sample is bypassed to avoid redundancy. Instead, another unique sample is selected. This process continues iteratively until the desired number of unique scenes, as specified, has been successfully compiled.

7 Implementation

This section details the implementation of 3D object detection using the BEVFusion model combined with a novel AL method, Active VisLED-Querying. The primary objective is to enhance detection performance by integrating diversity-based querying in an AL framework and comparing its efficacy against a baseline random querying method.

7.1 nuScenes Pre-processing

7.1.1 nuScenes Development Kit and File Setup Adjustments

The nuScenes dataset is sophisticated, encompassing a wide array of scenarios that make it ideal for 3D Object Detection applications. However, a notable limitation is the inflexibility of its underlying codebase, as it is hardcoded to work with all the scenes present, making it hard to use it with AL frameworks where not all scenes are required.

Consequently, the initial phase in adapting AL to the nuScenes dataset involved investigating the nuScenes development kit (nuScenes-devkit) [42], the primary codebase associated with the dataset. It was essential to identify and modify the specific code segments that restricted usage to the full dataset, enabling the use of dataset subsets instead. This modification was made in the 'splits.py' file, where it was possible to comment out an assertion line that specified the number of scenes required for the training, validation, and test splits individually. Subsequently, the hardcoded splits which contain the scene names of the available scenes, were disabled and replaced with empty lists.

To populate these empty lists, it was necessary to modify the JSON files located in the dataset folder structure. The nuScenes dataset organizes its JSON files into two directories: 'trainval', which contains all information pertaining to the training and validation sets, and 'test', which includes all information related to the test set. Notably, the test set lacks annotations because it is used exclusively for submissions to the nuScenes server, where results contribute to the nuScenes leaderboard rankings. Given that the 'test' split is not utilized in this implementation, it will be excluded from subsequent discussions.

As a diversity-based AL method that selects samples according to the diversity of image embeddings discussed in Section 6.1, there is no need to perform inference on the unused samples to identify the most diverse ones for subsequent rounds. This simplifies the file restructuring process, as no additional folders or files need to be created for these unused/unlabeled samples. Since all relevant information is tied to the scene names included in each split, the only necessary modification is to exclude the unused scenes from the file listing the scene names used in the training split.

These files will subsequently be linked to the splits.py script, and the scenes listed in the scene name files for each split will be added to the previously empty lists.

7.1.2 Image Embeddings

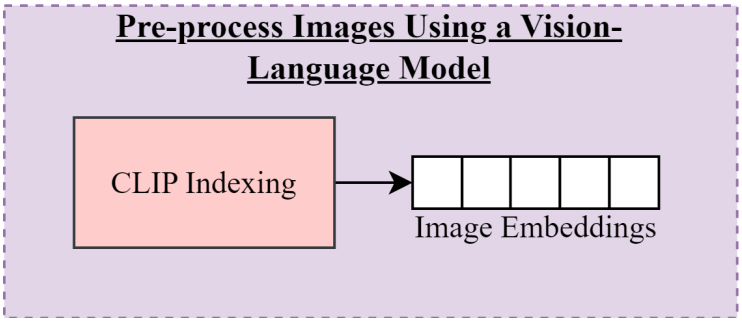


Figure 7.1. Creation of feature embeddings from the image inputs.

With the dataset now prepared for AL applications, the next step entails selecting the most diverse samples. To accomplish this, we will generate image embeddings through the image indexing function provided by the CLIP model, as shown in Figure 7.1. This will result in a matrix for each sample, which is then stored in a '.npy' file for clustering purposes. For Open-World scenarios, this procedure is performed once, while for Closed-World scenarios, it is conducted for each class individually.

7.2 Active VisLED-Querying

In the Closed-World application of the VisLED-Querying method, zero-shot learning is initially conducted on the classes. During this process, feature embeddings derived from the class names serve as language inputs, while feature embeddings derived from the sample images are utilized as image inputs. The classification of each image is then determined by calculating the cosine similarity between the image embeddings and the language embeddings. Each sample is classified into a single class, even if it contains multiple objects because the CLIP model only assigns high scores to the image's most prominent object. The objects achieving smaller scores are not considered, as the scores tend to be very close to each other percentage vise, this choice is made to avoid introducing false positives into the datasets.
While for both the Open-World Exploring and The Closed-World Mining methods the embeddings will be utilized as input to perform hierarchical clustering.

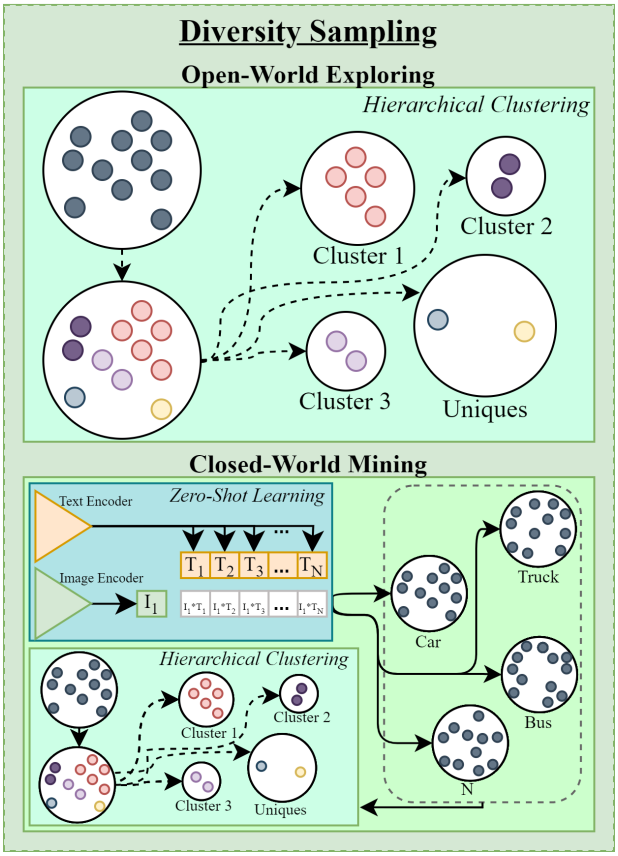


Figure 7.2. A representation of both Open-World and Closed-World VisLED-Querying.

The chosen method for clustering is the Agglomerative (bottom-up) approach, employing an average linkage technique. The number of clusters formed is variable and determined by a threshold specifically selected to optimally represent the data. Upon selecting the clusters, the images associated with each matrix will be organized into a new folder, each folder representing one cluster, with the exception of the unique samples which do not fit into any cluster, they will be gathered into a single folder.

In the Open-World Exploring approach, clustering is performed on the entire dataset at once, whereas in the Closed-World Mining approach, clustering is carried out independently for each class generated by the zero-shot learning phase.

The algorithm will then proceed to select samples based on the steps outlined in Section 6.3. The Open-World Exploring technique, will choose from a singular 'unique' directory containing all of the most diverse samples in the training set. In contrast, the Closed-World Mining technique will involve sampling an equal amount of diverse samples from 'unique' directories established for each class during the clustering phase.

7.3 BEVFusion

When implementing BEVFusion, the configuration file required modifications to fit the constraints of the existing hardware. For this project, only one Nvidia RTX 4090 GPU was

available, unlike the original authors of the study, who suggested using eight GPUs [43]. This necessitated changes to both the batch size and the learning rate. Given the memory limitations of the single GPU, which has 24 GB of graphics RAM [44], the batch size was reduced to 1 to avoid depleting GPU memory with larger batch sizes.

The comprehensive architecture of the entire system is depicted in Figure 7.3. This configuration is achieved by integrating various components of the setup into a single script that automates the process. This script enables the setup of the dataset, as well as its training and testing using the train and test scripts supplied by the developers of the BEVFusion model, for each specified dataset split.

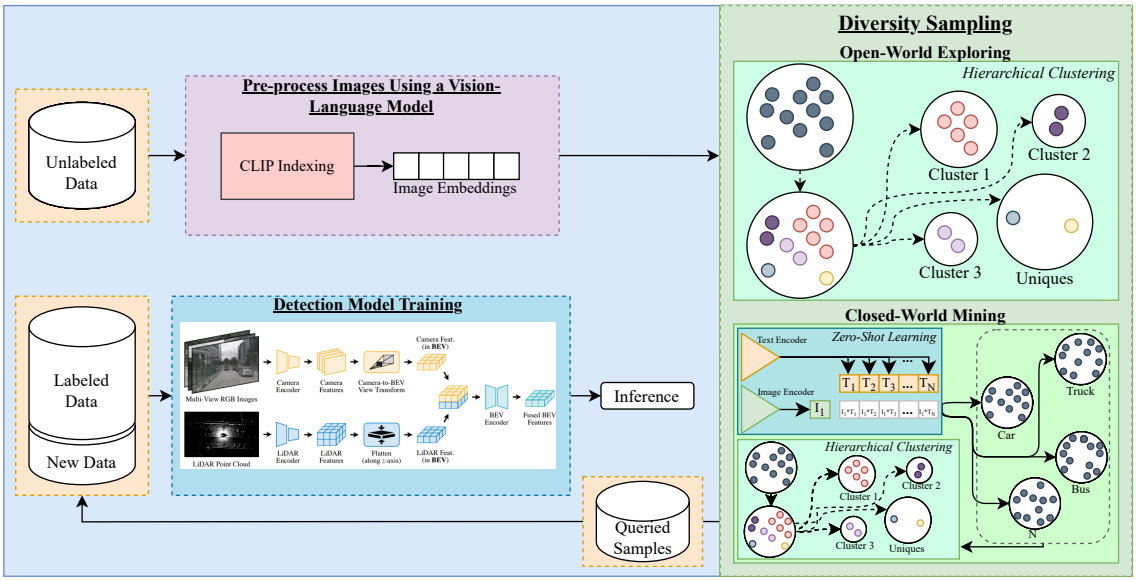


Figure 7.3. Training pipeline. (Created for the paper seen in Appendix B)

To effectively save and manage the output results generated by the scripts, we utilize the *Weights & Biases* platform. This platform provides robust tracking and visualization tools, enabling seamless logging of metrics, system parameters, and model outputs. By integrating *Weights & Biases*, we ensure that all experimental data is stored in an organized manner, facilitating easy access and analysis [45].

Testing 8

The testing chapter serves to evaluate the effectiveness and robustness of the proposed VisLED-Querying method for 3D object detection. This section is dedicated to outlining the setup and results of the experimental testing conducted to validate the performance of the developed algorithms.

In the subsequent sections, the specifics of the experimental setup will be delved into, while presenting quantitative results derived from various performance metrics, and offering a qualitative analysis of the findings. The aim is to demonstrate the efficacy of VisLED-Querying in achieving high performance with reduced data, thereby underscoring the potential benefits of AL approaches in the realm of 3D object detection.

8.1 Experimental Setup

Initially, only 10 % of the dataset will be utilized in the first iteration, with each subsequent iteration incorporating an incremental 10 % of the data, meaning that 100 scenes are added each iteration until 50 % of the dataset is employed. This incremental training approach will yield five distinct models for each sampling technique, with each model undergoing training across six epochs. Since training five models with six epochs takes approximately six days, a higher number of epochs, although potentially yielding better results, was not chosen. This decision ensures sufficient time to conduct all tests. This progression from the initial 10 % to the final phase using 50 % of the dataset allows for a comprehensive assessment of the model's learning capabilities and the effectiveness of each sampling method under different data volumes.

The VisLED-Querying technique will be executed five times for each dataset split to establish that the algorithm delivers consistently strong performance. This repeated testing is crucial to statistically validate the reliability and effectiveness of the VisLED-Querying method. However, due to time constraints, the experiment using the Random Sampling method will be conducted only once. As a result, the outcomes presented for the VisLED-Querying method will consistently include both the mean and the standard deviation, providing a more detailed insight into its performance. Where on the other hand, the results from the Random Sampling method will be less comprehensive.

For the VisLED-Querying Closed-World Mining method, a decision was made to exclude the three most frequently occurring classes (car, pedestrian, and temporary traffic barrier) from the zero-shot learning sampling, focusing instead on the remaining seven classes. This decision was made due to the disproportionately large amount of data available for these classes. Since each image is classified into a single class to identify the most prominent

object, most samples would otherwise be classified into one of these common classes, even if they contain objects from the rarer classes.

8.1.1 Architecture Limitations

The availability of only one GPU for this project imposed certain restrictions on both the algorithm and the model.

Due to an insufficient GPU size, the computer was unable to execute hierarchical clustering on all image embeddings simultaneously. Consequently, the Open-World exploring method is restricted to handling only 19,000 images. These images are selected randomly from the dataset. This specific number was determined through a process of elimination, which established that clustering 19,000 images consistently proceeds without errors. However, attempting to cluster more often leads to frequent 'out of memory' errors during the process. The reason is that the GPU memory required surpasses the amount of data the GPU can accommodate simultaneously, as many complex calculations are happening, for the clusters to be accurate.

For this project, it was crucial to consider that the original implementation of the BEVFusion architecture utilized 8 GPUs, a setup that significantly exceeds our hardware capabilities. To adapt the architecture to the single GPU available for this project, various hyperparameters required adjustments. The most critical changes were made to the learning rate and the batch size to accommodate the reduced computational power. Additionally, the 'workers' parameter had to be modified, as processing attempts consistently failed when attempting to handle more than 40 % of the dataset. This issue arose from BEVFusion's limitations in managing multiprocessing tasks with limited GPU resources. The adjustments to these parameters were necessary to ensure the model's operational stability and performance on the limited hardware. A comparison between the original hyperparameters and those applied in this project is detailed in Table 8.1.

Hyperparameters	Original	Changes
Learning Rate	2.0e-4	2.0e-5
Batch size	6	1
Workers	4	1
Epochs	6	10

Table 8.1. Overview of changes made to the hyperparameters of the model.

8.1.2 Hierarchical Clustering Threshold Placement Test

Method:

To determine the appropriate threshold for hierarchical clustering, it is essential to test various thresholds. This experiment involves examining the individual clusters formed and assessing their sizes. Additionally, the unique folder will be inspected to identify any images that should be included in a cluster.

The process begins by selecting a low threshold value and gradually increasing it until false positives start to appear in the clusters. The final threshold will be the value just before false positives are observed.

Results:

The initial threshold chosen was 0.2. Upon observation, it was found that this threshold was too low, as there were barely any images in the clusters, with the majority of images being located in the 'unique' folder.

The next threshold tested was 0.3, which resulted in better cluster formation. A pattern began to emerge, with e.g. images taken in the dark with no objects present often clustered together and street corners that the vehicle had passed multiple times, showing different lighting conditions, were also grouped together. However, many images that should have been part of clusters remained in the 'unique' folder. This was easily observable due to the ego vehicle stopping at crossings and red lights, resulting in many similar samples that were easily identifiable while located in the 'unique' folder.

The same pattern persisted when testing thresholds of 0.4 and 0.45. Consequently, smaller increments were chosen as the 'unique' folder started to become more diverse but was still not sufficiently accurate. Testing the 0.5 threshold improved the diversity of the 'unique' folder without introducing false positives into the clusters. Higher thresholds were also tested, but the clusters began to include false positives.

Therefore, a threshold of 0.5 was selected for all further experiments.

8.1.3 Zero-Shot Learning Test

Method:

To conduct this test, all classes will first be converted into language embeddings. Then, the cosine similarity will be calculated for each sample's image embedding, and the sample will be assigned to a class. Once this process is completed for the entire dataset, observations will be made on the number of images fitting into each class and the quality of the selections for each class. This will involve checking the images to determine if an object from the chosen class is present.

Results:

Initially, language feature embeddings were created for all 10 classes. However, when categorizing the images based on all classes, it was noticed that the majority of images ended up in the Car, Pedestrian, or Temporary Traffic Barrier classes. This is because Cars and Pedestrians are common on most roads, often resulting in the highest accuracy for these classes. Additionally, the CLIP model, trained on highly diverse data, tends to classify any type of barrier under the Temporary Traffic Barrier class if it has the highest accuracy in the sample. Observations of the Temporary Traffic Barrier class revealed the inclusion of fences, walls, and even some hills.

Since the algorithm is designed to facilitate the identification of less represented classes, and given the high number of false positives in the Temporary Traffic Barrier class, these classes were omitted from further testing.

When testing the seven remaining classes, it was found that the categorization of images better represented the objects present in each image. However, there were still false positives

included in the classes. e.g, in the bicycle class, some images included only bicycle racks without bicycles. Similarly, in the bus class, bus stops without buses were included. This is again due to the diversity of the data on which the CLIP model is trained and the broader associations of the words used in categorization.

Despite these issues, it was decided to keep these classes because they are underrepresented in the dataset and therefore important to include in the sampling process.

8.2 Quantitative Results

8.2.1 VisLED-Querying vs. Random Sampling

Method:

The experimental results presented here correspond to the procedures outlined in Section 8.1.

The results will be displayed for the random sampling method alongside the two VisLED methods, which are compared to the outcomes obtained from the full dataset run without AL.

Result:

Table 8.2 reveals that both VisLED-Querying scenarios yield higher accuracies than Random Sampling. Furthermore, the Open-World Exploring scenario surpasses the Closed-World Mining scenario in accuracy, although the Closed-World Mining scenario shows a lower standard deviation, indicating more consistent performance.

Additionally, it can be seen that the standard deviation for the Closed-World Mining scenario generally decreases with each iteration as the labeled training set expands. However, no consistent pattern is observed in the standard deviation fluctuations for the Open-World Exploring scenario.

When examining the mAP score, the Open-World Exploring and Closed-World Mining scenarios show a reduction in accuracy of 1.28 % and 1.93 %, respectively, compared to training on the complete dataset without AL. Meanwhile, Random Sampling leads to a 2.98 % lower accuracy.

In terms of the NDS score, the Open-World Exploring and Closed-World Mining scenarios have accuracies that are 2.1 % and 2.37 % lower, respectively, compared to the NDS score of the full training set without AL. Meanwhile, the Random Sampling method results in a 3.09 % lower accuracy.

Labeled Pool		mAP					NDS				
Rounds	%	Random	VisLED (CWM)		VisLED (OWE)		Random	VisLED (CWM)		VisLED (OWE)	
			Mean	STD	Mean	STD		Mean	STD	Mean	STD
1	10%	30.95	28.98	0.34	32.02	0.71	33.53	32.54	0.31	34.76	0.65
2	20%	38.00	40.92	1.09	41.68	0.82	40.14	41.54	0.65	42.78	1.13
3	30%	44.94	45.44	0.89	46.81	0.37	48.41	48.93	0.79	50.99	1.05
4	40%	47.73	49.35	0.51	49.52	0.59	53.10	53.54	0.35	54.89	0.52
5	50%	**49.90**	**50.95**	0.14	**51.60**	0.98	**55.64**	**56.36**	0.36	**56.63**	0.94
	100%	**52.88**					**58.73**				

Table 8.2. This table shows the mean average precision (mAP) and nuScenes detection score (NDS) metrics for the random sampling, and VisLED-querying (Closed-World Mining (CWM) and Open-World Exploring (OWE)) in every round. It also shows the mAP and NDS scores for the full training split when trained using one GPU.

8.2.2 Class Distribution Comparison

Method:

To visualize the efficiency of the VisLED-Querying Method, the class distribution of all objects included in the training will be extracted from the wandb log files generated during each run.

Results:

Figure A.1 demonstrates that the VisLED-Querying method effectively samples more diverse data across most splits. For classes with limited data, the VisLED-Querying method tends to sample more data in these instances. Additionally, the Closed-World algorithm specifically manages to sample less data from the more prevalent classes, such as pedestrian, temporary traffic barrier, truck, and traffic cone. Plots comparing the class distribution to the full training set can be seen in Appendix A.

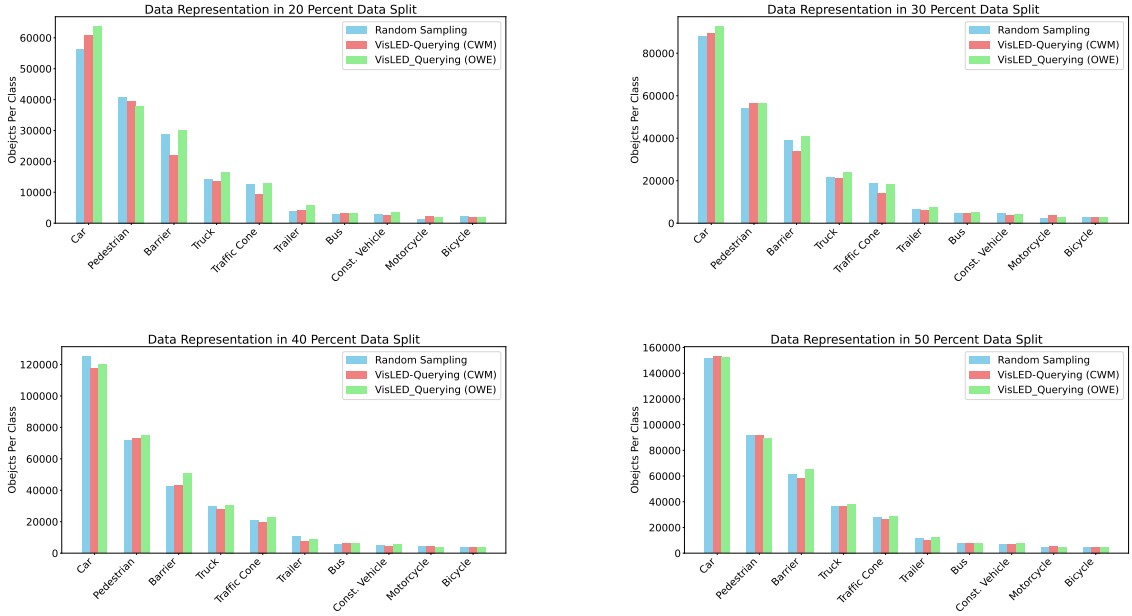


Figure 8.1. Class distribution between Random Sampling, VisLED-Querying Open-World Exploring (OWE) and Closed-World Mining (CWM).

8.2.3 Class Comparison: VisLED-Querying Closed-World Mining vs. Random Sampling

Method:

In this analysis of the conducted experiment, the mean Average Precision (mAP) for each class will be derived from the models. Following this, a detailed examination of the classes will be conducted to assess whether VisLED-querying achieves its intended effect. Initially, a summary of class performances will be presented, which will facilitate an individual analysis of each class. This analysis will determine which classes underperformed and whether they exhibited improved performance via VisLED querying compared to random sampling.

Furthermore, an evaluation of the graphical data will be complemented by an analysis of the sample distribution per class, as indicated in Tables 3.6 and 3.7. These tables provide a clearer picture of the incremental samples added in each sampling round. Should the test prove successful, there will be an increase in the data for classes demonstrating the lowest accuracy, as depicted in the tables and illustrated in the figures. It is important to note, however, that while sampling occurs at the scene level, all classes will see an increase in data with each round. Nonetheless, the data augmentation should be more pronounced in classes with fewer samples as these are most likely to contain diverse samples.

Results

The results presented in Tables 8.3 and 8.4 indicate that VisLED-Querying outperforms Random Sampling for most classes, as also illustrated in Figure 8.2. Additionally, it is evident that for the majority of classes, using 50 % of the data yields accuracy levels close to when using 100 % of the data. An exception is the Truck class, seen in Figure 8.4d which significantly exceeds the accuracy achieved with the full data set and the Cicycle class, seen in Figure 8.4b which yields lower accuracies. To thoroughly examine each class, Figure 8.2 is divided to allow for detailed analysis of individual classes.

Labeled Pool		Car		Truck		Bus		Const. Vehicle		Motorcycle	
Rounds	%	Random	VisLED	Random	VisLED	Random	VisLED	Random	VisLED	Random	VisLED
1	10%	74.54	71.88	29.56	25.04	23.01	21.79	6.97	5.24	21.45	17.08
2	20%	78.41	78.72	33.83	35.54	36.95	38.00	8.32	11.97	22.92	36.98
3	30%	81.72	81.29	31.40	40.20	46.41	42.69	16.43	13.95	37.99	45.42
4	40%	83.57	83.44	36.87	39.93	49.66	51.20	15.44	19.08	48.22	52.20
5	50%	**84.00**	**84.46**	**37.49**	**46.48**	**55.26**	**52.29**	**18.46**	**20.89**	**52.33**	**53.21**
	100%	**85.24**		**39.27**		**54.45**		**22.25**		**56.30**	

Table 8.3. The mAP score for the classes are shown, for Random Sampling and VisLED (Closed-World Mining) where the VisLED results represent a mean value of five runs, while the results for the Random Sampling method only represent one run.

Labeled Pool		Bicycle		Trailer		Pedest.		Traff. Cone		Traff. Barrier	
Rounds	%	Random	VisLED	Random	VisLED	Random	VisLED	Random	VisLED	Random	VisLED
1	10%	7.42	2.71	2.02	5.07	58.34	55.22	53.62	50.08	32.14	35.63
2	20%	10.41	18.07	8.13	11.52	68.00	62.95	61.10	60.04	51.93	55.04
3	30%	18.47	19.10	15.57	15.08	77.04	77.01	65.95	63.84	**58.47**	55.77
4	40%	21.50	**26.18**	17.52	**18.93**	79.55	80.17	66.90	66.03	58.06	56.15
5	50%	**26.28**	24.60	**18.91**	18.92	**80.08**	**81.64**	**68.75**	**69.07**	57.49	**57.92**
	100%	**32.67**		**23.17**		**83.54**		**70.32**		**66.04**	

Table 8.4. The mAP score for the classes are shown, for Random Sampling and VisLED (Closed-World Mining) where the VisLED results represent a mean value of five runs, while the results for the Random Sampling method only represent one run.

As illustrated in Figure 8.2, the three classes excluded from the zero-shot learning component (Car, Pedestrian, and Temporary Traffic Barrier), exhibit nearly identical performance curves for both Random Sampling and VisLED-Querying, as further explored in Figure 8.3. Additionally, Tables 8.3 and 8.4 show that the accuracy for the Car and Pedestrian classes as being 0.78 % and 1.9 % lower, respectively compared to the results from the full dataset. In contrast, the accuracy for the Temporary Traffic Barrier class is 8.12 % lower than the full dataset results but still higher than the accuracy achieved through Random Sampling.

Additionally, Figure 8.2 and Table 8.4 show that the Traffic Cone class achieves high accuracy, despite representing only 8.40 % of the full dataset, as indicated in Table 5.2. therefore, it was not taken into consideration when choosing which class to exclude from the zero-shot learning classifications.

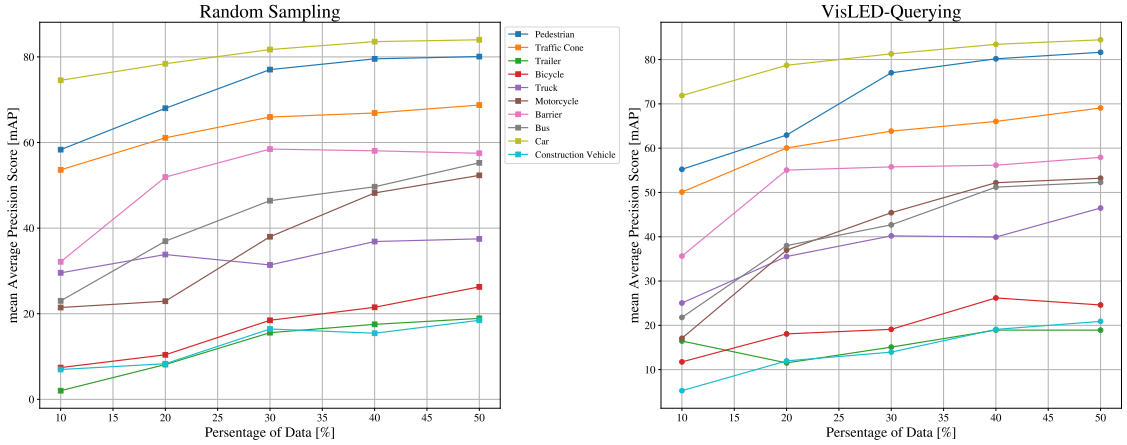


Figure 8.2. Results for all classes split into Random sampling and VisLED-Querying (Closed-World Mining).

Furthermore, Figure 8.2 reveals that the accuracy of the remaining classes varies significantly. Some classes, such as Truck, Motorcycle, and Bus, maintain good accuracy, although not as good as the classes with a lot of data present, while others, like Bicycle, Trailer, and Construction Vehicle, perform poorly. Although there is variation, most classes perform better with VisLED-Querying than with Random Sampling. Some classes show only slight improvement, while others maintain significantly higher accuracy, thereby reinforcing the credibility of the developed AL method.

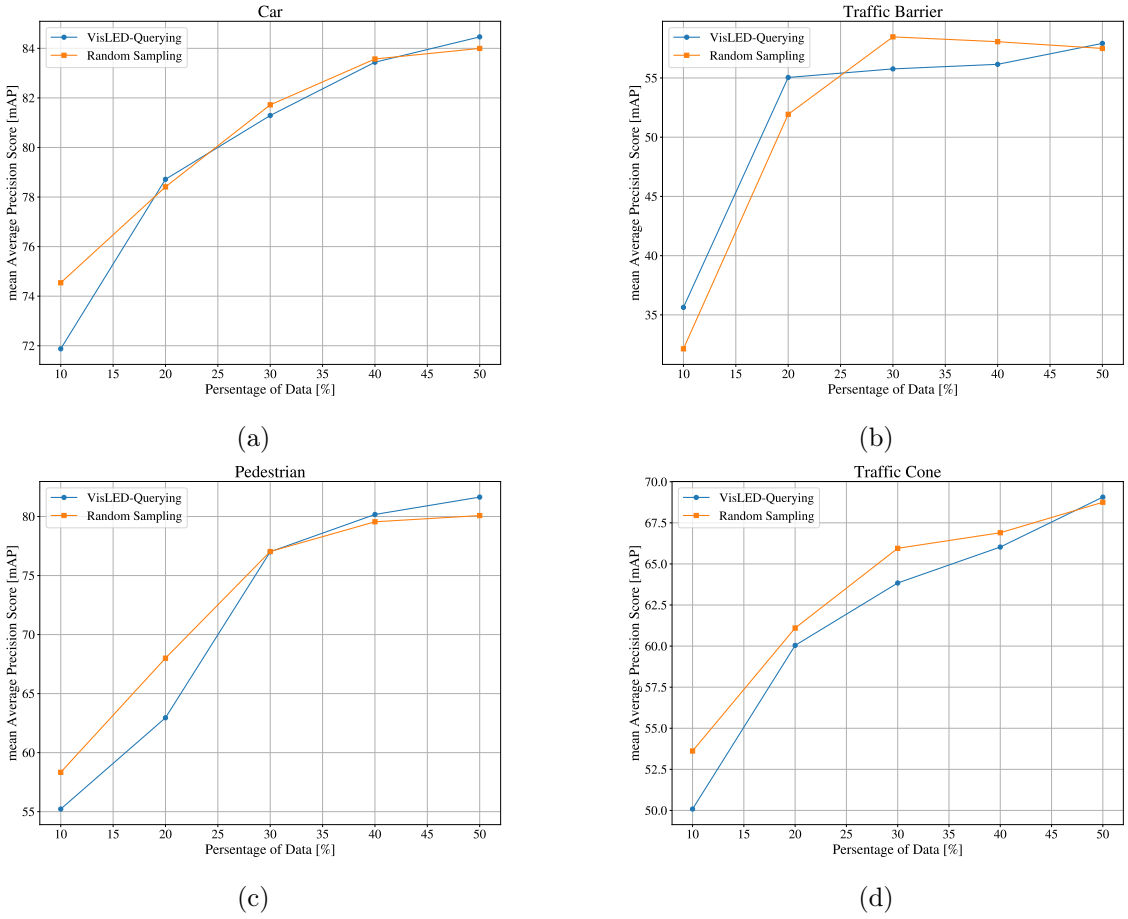


Figure 8.3. Representation of the stability of the most frequently occurring classes.

One particularly interesting class to highlight is the Truck class. When using 50 % of the data, this class exceeds the accuracy achieved with the full training set by 7.21 % and outperforms Random Sampling by 8.99 %, as shown in Table 8.3. These results can also be observed in Figure 8.4d.

Several classes do not achieve high accuracies even with the full training set, as seen in Tables 8.3 and 8.4, with the Motorcycle class being one example. Nevertheless, a significant achievement is noted in the Motorcycle class. Figure 8.4e shows that the VisLED-Querying method consistently maintains significantly higher accuracy compared to the Random Sampling method, although the accuracy gap narrows by the final iteration.

Another such class is the Construction Vehicle class, as illustrated in Figure 8.4f. Where the VisLED-Querying method generally outperforms the Random Sampling method, with the only exception being the 30 % data split.

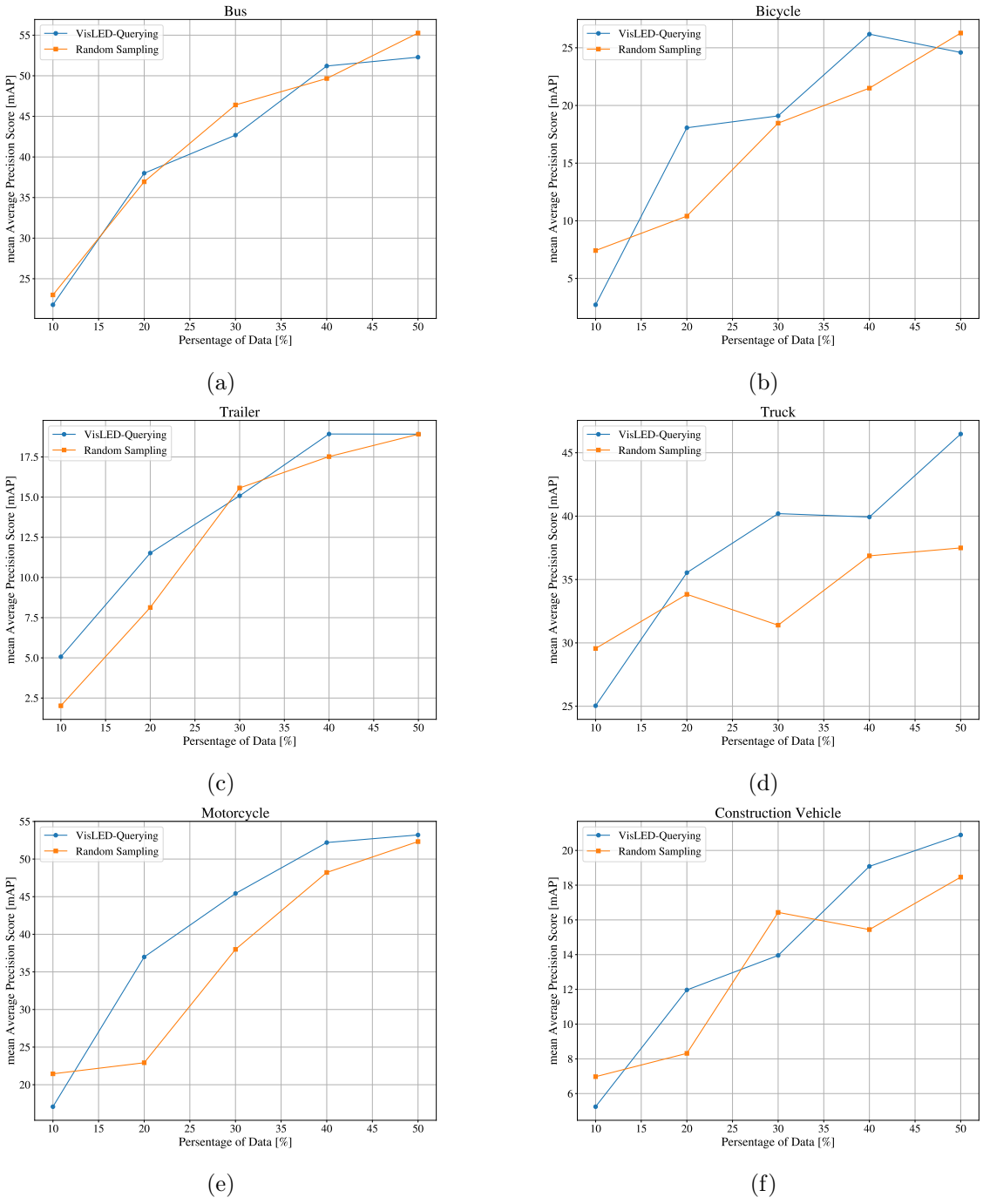


Figure 8.4. mAP score of other classes.

8.2.4 Class Comparison: VisLED-Querying Open-World Exploring vs. Random Sampling

Method:

This analysis follows the same principles as the prior class comparison analysis seen in Section 8.2.3 for VisLED-querying Closed-World Mining.

Results:

Tables 8.5 and 8.6 show that the accuracies for each class in the Open-World Exploring scenario are generally higher compared to those from Random Sampling. This trend holds for all classes except for the Motorcycle, Bicycle and Bus classes, which achieve lower accuracies with the Open-World Exploring method than with Random Sampling.

Additionally, it is noted that three classes (Truck, Construction Vehicle, and Traffic Cone) outperform the accuracy achieved when using the full dataset without AL for training.

Labeled Pool		Car		Truck		Bus		Const. Vehicle		Motorcycle	
Rounds	%	Random	VisLED	Random	VisLED	Random	VisLED	Random	VisLED	Random	VisLED
1	10%	74.54	74.33	29.56	28.85	23.01	27.45	6.97	5.22	21.45	23.16
2	20%	78.41	79.64	33.83	35.36	36.95	40.32	8.32	15.33	22.92	35.47
3	30%	81.72	81.97	31.40	42.11	46.41	47.11	16.43	17.69	37.99	47.53
4	40%	83.57	83.74	36.87	41.44	49.66	51.64	15.44	22.07	48.22	46.00
5	50%	**84.00**	**84.55**	**37.49**	**42.36**	**55.26**	**53.68**	**18.46**	**22.36**	**52.33**	**50.91**
	100%	**85.24**		**39.27**		**54.45**		**22.25**		**56.30**	

Table 8.5. The mAP score for the classes are shown, for Random Sampling and VisLED (Open-World Exploring) where the VisLED results represent a mean value of five runs, while the results for the Random Sampling method only represent one run.

Labeled Pool		Bicycle		Trailer		Pedest.		Traff. Cone		Traff. Barrier	
Rounds	%	Random	VisLED	Random	VisLED	Random	VisLED	Random	VisLED	Random	VisLED
1	10%	7.42	1.58	2.02	9.10	58.34	57.82	53.62	51.34	32.14	41.29
2	20%	10.41	12.51	8.13	17.95	68.00	64.77	61.10	65.01	51.93	50.49
3	30%	18.47	18.68	15.57	14.20	77.04	77.50	65.95	66.75	**58.47**	54.55
4	40%	21.50	20.38	17.52	19.93	79.55	78.84	66.90	68.12	58.06	63.04
5	50%	**26.28**	**24.54**	**18.91**	**22.25**	**80.08**	**81.56**	**68.75**	**70.63**	57.49	**63.20**
	100%	**32.67**		**23.17**		**83.54**		**70.32**		**66.04**	

Table 8.6. The mAP score for the classes are shown, for Random Sampling and VisLED (Open-World Exploring) where the VisLED results represent a mean value of five runs, while the results for the Random Sampling method only represent one run.

Figure 8.5 illustrates that the distribution of classes across the mAP score is similar to the distribution observed in Section 8.2.3. In both cases, classes with more data achieve higher accuracies than those with less data.

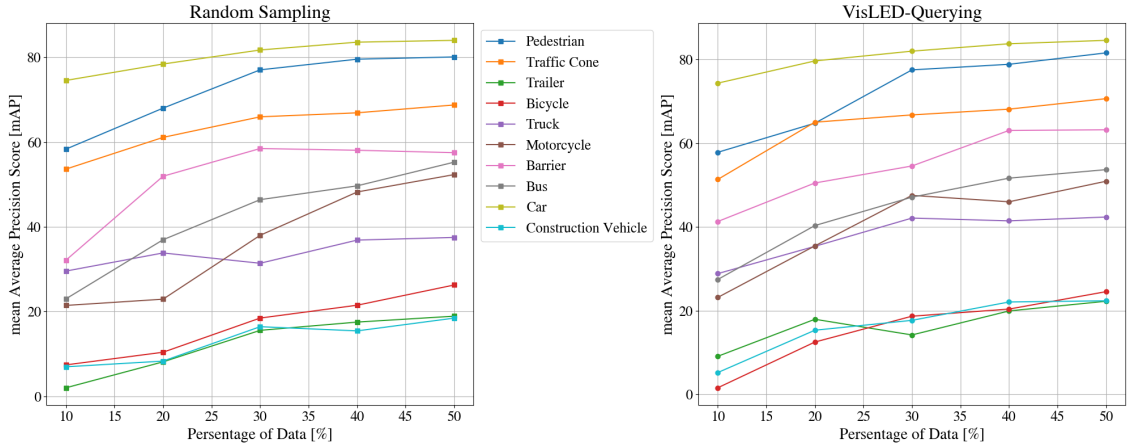


Figure 8.5. Results for all classes split into Random sampling and VisLED-Querying (Open-World Exploring).

This pattern is also evident in Figure 8.6, where the graphs follow a similar curve to the previous experiment. However, the VisLED-Querying method attains the highest accuracy for all classes. Furthermore, for the Traffic Cone class, as seen in Figure 8.3d, VisLED-Querying achieves a higher accuracy than the full dataset, as presented in Table 8.6.

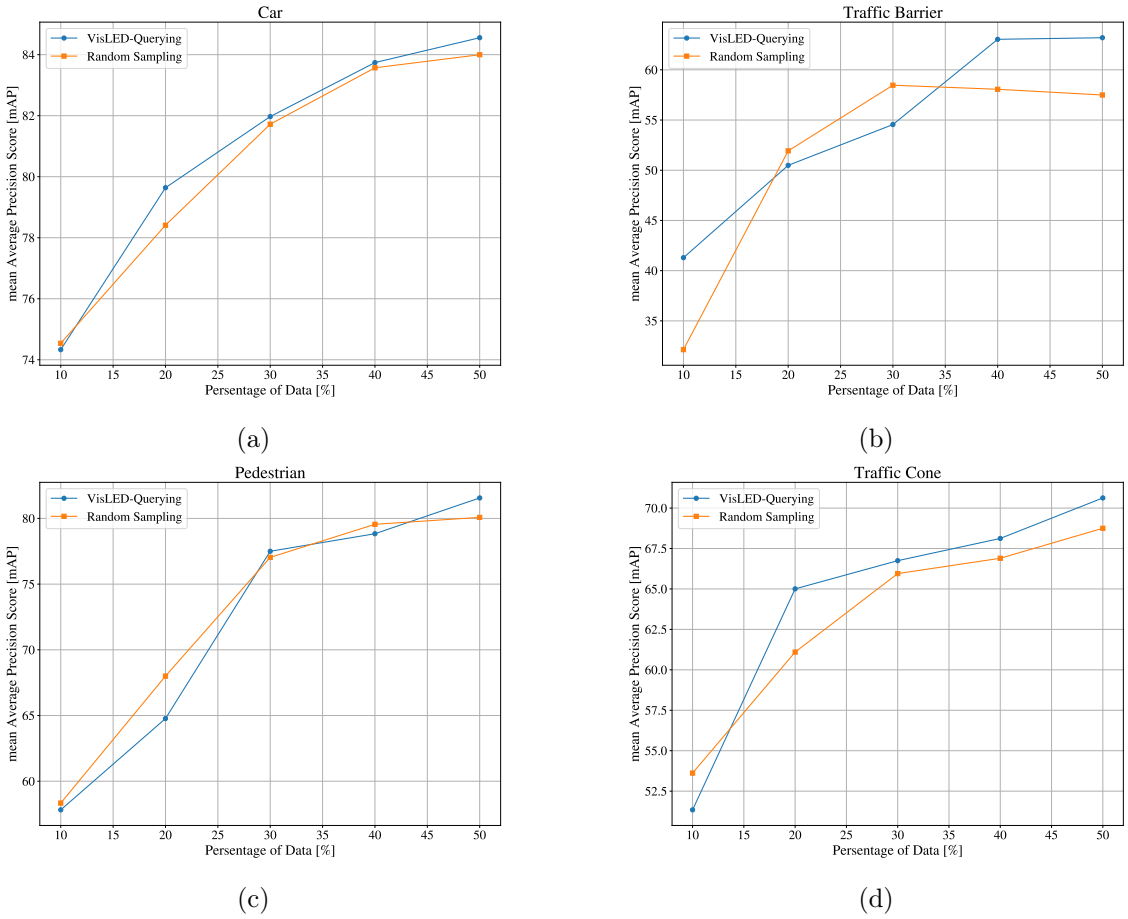


Figure 8.6. Representation of the stability of the most frequently occurring classes.

The classes show varying results with this method. Random Sampling achieves better performance than the VisLED-Querying Open-World Exploring method for the Bus class (Figure 8.7a), Bicycle class (Figure 8.7b), and Motorcycle class (Figure 8.7e). Conversely, the VisLED-Querying method excels in seven classes, three of which also surpass the accuracy of the fully trained dataset.

Notably, two classes that outperform the fully trained dataset without AL have relatively few data points present in the labeled dataset. These are the Truck and Construction Vehicle classes, as shown in Figures 8.7d and 8.7f, respectively, and in Table 8.5.

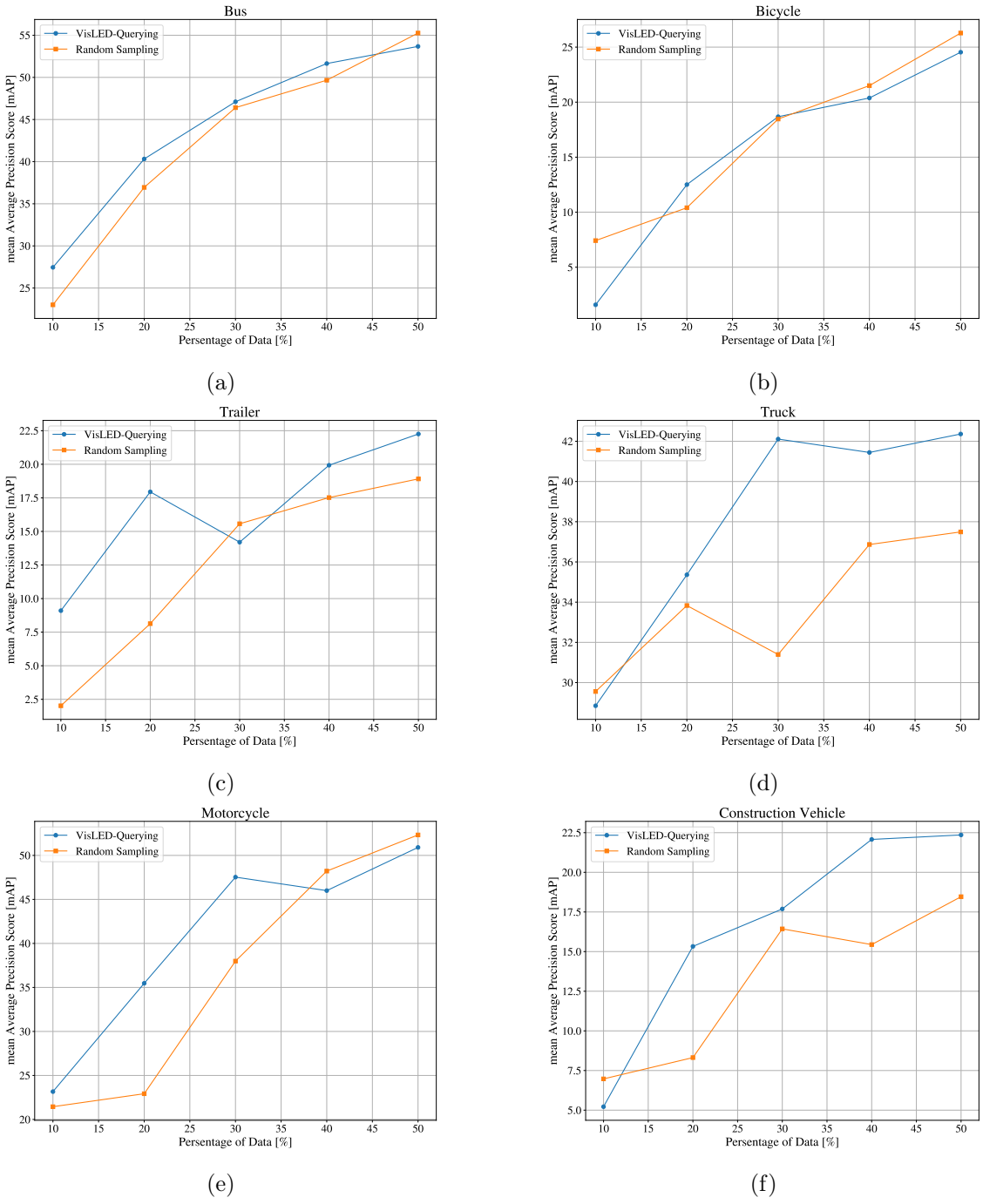


Figure 8.7. mAP score of other classes.

8.3 Qualitative Analysis

8.3.1 Sample Evaluation

Method:

For this qualitative analysis, 10 different images will be randomly chosen to represent a diverse range of objects and weather conditions, highlighting the effectiveness of each VisLED method. The results will be shown for 50 % VisLED-Querying Closed-World Mining, 50 % VisLED-Querying Open-World Exploring, and compared with 50 % Random Sampling, the full training set without AL, and the ground truth for each image.
This approach aims to provide a comprehensive evaluation of the VisLED-Querying methods under varied conditions. By making this comparison, the analysis will demonstrate the strengths and limitations of each method. The inclusion of diverse weather conditions and objects in the selected images will ensure that the evaluation covers a broad spectrum of real-world scenarios, providing insights into the practical applicability of the VisLED-Querying methods.

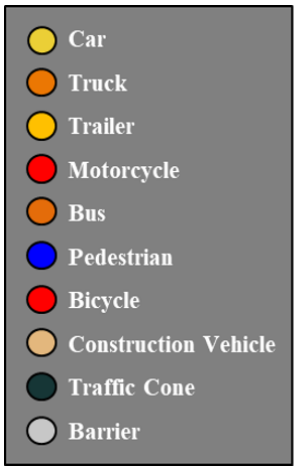


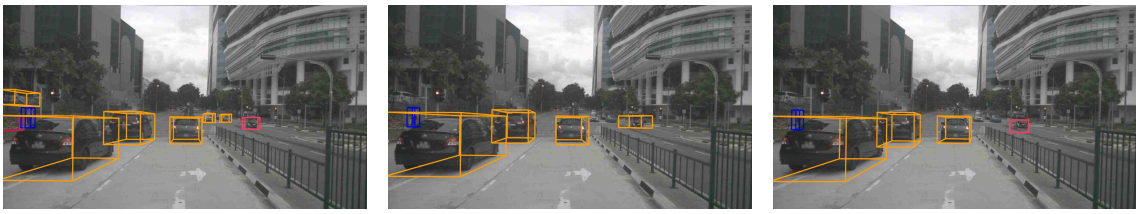
Figure 8.8. Colors of each class in the visualization of the classes.

In each sample, the bounding boxes will be visible in different colors, representing the individual classes. The colors corresponding to each class can be seen in Figure 8.8.

Results:

In the first sample selected, objects such as cars, a pedestrian, a motorcycle, and a bicycle are visible in cloudy weather conditions. These are all highlighted in Figure 8.9a. It is evident that all the algorithms can detect the three closest cars and the pedestrian, but they struggle to detect the cars that are furthest away and the bicycle.

Figure 8.9b shows that the Random Sampling method incorrectly classifies the motorcycle as a car, while the full training set, seen in Figure 8.9e fails to detect the motorcycle altogether. In contrast, both VisLED-Querying methods, illustrated in Figures 8.9c and 8.9d, accurately classify the motorcycle. This demonstrates the enhanced capability of VisLED-Querying methods in recognizing and correctly classifying objects, particularly in complex scenes with multiple object types.



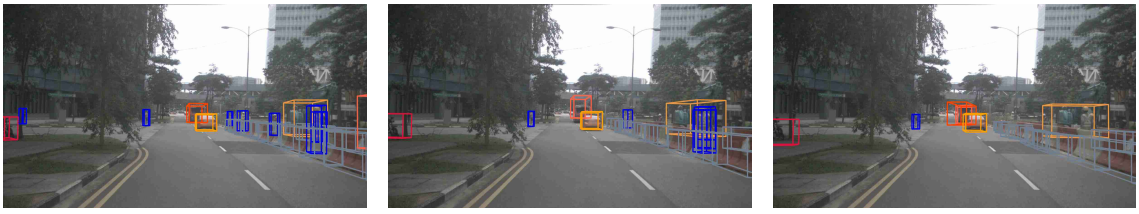
(a) Ground Truth (b) Random (c) Closed-World Mining

(d) Open-World Exploring (e) Full Training set (No AL)

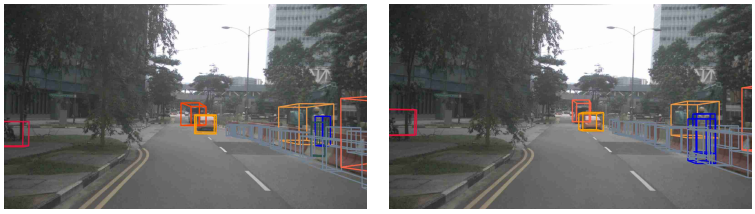
Figure 8.9. A sample with objects including Car, Pedestrian, Motorcycle, and Bicycle.

In the sample provided in Figure 8.10a, various objects are labeled, including a car, a bicycle, pedestrians, a bus, temporary traffic barriers, a construction vehicle, and a truck in cloudy weather conditions. All methods correctly label the car, bicycle, and barriers. They also accurately classify the bus and construction vehicle, albeit with varying levels of precision, as indicated by the differences in bounding box accuracies.

The methods demonstrate proficiency in identifying different pedestrians, although none can detect all of them. Notably, the VisLED-Querying Open-World Exploring method, as shown in Figure 8.10d, and the full training set, as seen in Figure 8.10e, manage to identify the truck in the right corner of the image, despite only a small portion of it being visible.



(a) Ground Truth (b) Random (c) Closed-World Mining



(d) Open-World Exploring (e) Full Training set (No AL)

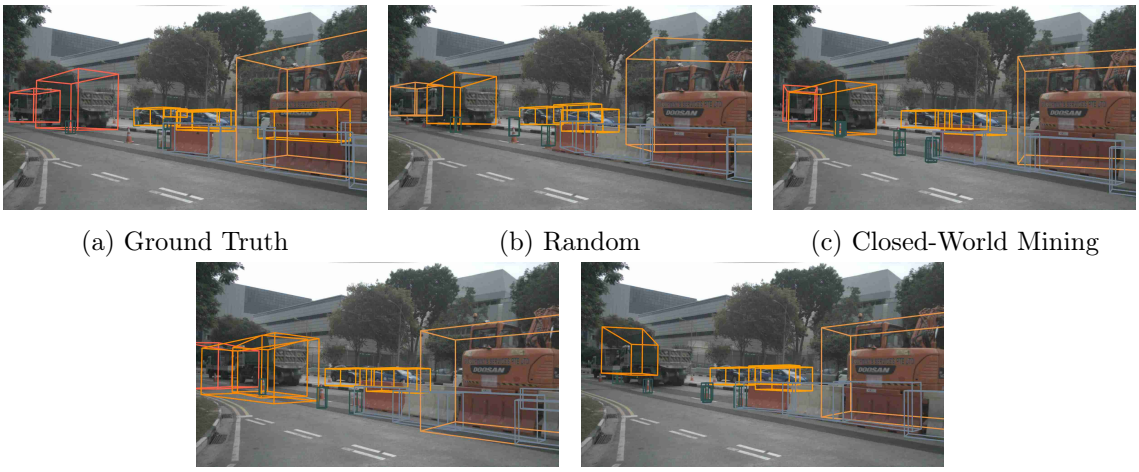
Figure 8.10. A sample with objects including Car, Pedestrian, Bicycle, Barrier, Construction Vehicle, Truck and Bus.

In the sample shown in Figure 8.11a, the objects visible include, cars, temporary traffic barriers, traffic cones, trucks, and a construction vehicle, all in a cloudy environment. It is observed that the temporary traffic barriers and easily identifiable cars are correctly

labeled by all methods. However, the car that is barely visible is not identified by any of the methods. The construction vehicle is correctly identified by all methods, but Random Sampling, shown in Figure 8.11b, has an incorrect bounding box rotation.

Although not all traffic cones are annotated in Figure 8.11a, the methods still correctly identify the unannotated traffic cone, albeit with varying precision.

The two trucks seen furthest away in the sample, are not accurately identified by any of the methods. In Figure 8.11b, the Random Sampling method incorrectly classifies the trucks as cars. Figure 8.11c shows that the VisLED-Querying Closed-World Mining method correctly identifies the truck furthest away but misclassifies the closer truck as a car. Similarly, in Figure 8.11d, the VisLED-Querying Open-World Exploring method makes the same mistake but also includes an additional bounding box, likely misidentifying a third vehicle in the area. The results for the full training set, seen in Figure 8.11e, show only one identified object, which is misclassified as a construction vehicle instead of a truck, and the bounding box orientation does not align with the actual object.



(a) Ground Truth (b) Random (c) Closed-World Mining

(d) Open-World Exploring (e) Full Training set (No AL)

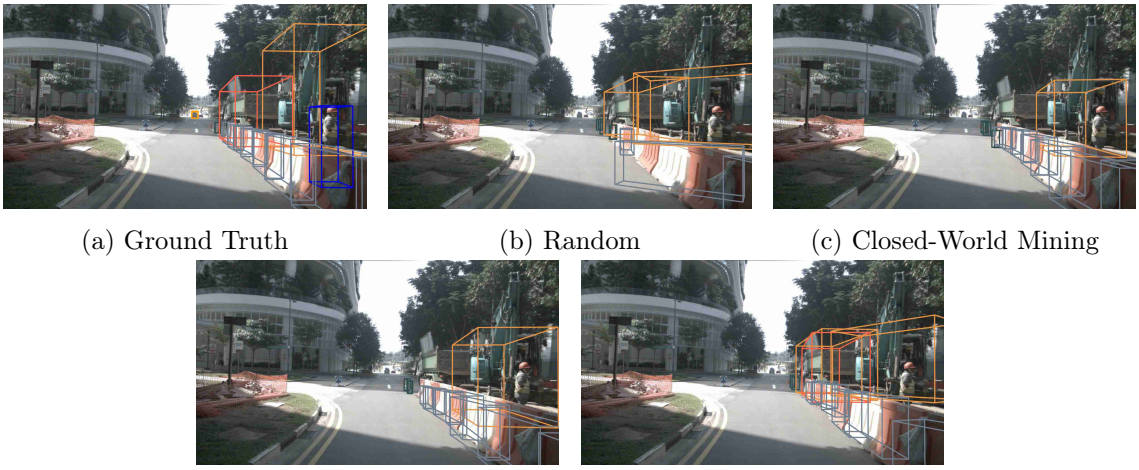
Figure 8.11. A sample with objects including Car, Truck, Construction Vehicle, Traffic Cone and Barrier.

In the next sample, observed in Figure 8.12a, objects such as a car, temporary traffic barriers, a truck, a pedestrian, and traffic cones are visible in cloudy conditions with the sun shining in from the side. All methods correctly identify the construction vehicle. However, the Random Sampling method, seen in Figure 8.12b, and the full training set, observed in Figure 8.12e, produce bounding boxes that are incorrectly oriented and too small.

The truck is accurately identified only by the full training set, but it also includes an additional bounding box misclassifying the truck as a car. The car in the distance and the pedestrian are not correctly identified by any of the methods.

All methods, except Random Sampling, correctly identify the temporary traffic barriers. The Random Sampling method is clearly confused by the sun. Additionally, the traffic cones are correctly labeled by the methods. However, both the Random Sampling method and the VisLED-Querying Closed-World Mining method, seen in Figure 8.12c, wrongly

identify a traffic cone where the sun hits the temporary traffic barriers. This likely occurs because the objects are similar in color, and the drastic color change created by the part of the temporary traffic barrier that is in shadow and the part that is in the sun, creates an object with two different lighting conditions. This makes it appear as if there are two distinct objects, increasing the likelihood that the algorithms will misidentify part of the temporary traffic barrier as a traffic cone.



(a) Ground Truth (b) Random (c) Closed-World Mining

(d) Open-World Exploring (e) Full Training set (No AL)

Figure 8.12. A sample with objects including Car, Pedestrian, Truck, Barrier, Traffic Cone and Construction Vehicle.

In the sample, seen in Figure 8.13a, only truck and trailer objects are present. The lighting conditions are quite challenging, as the image is dark due to cloudy weather with light shining through intermittently. It is evident that none of the methods correctly identify the truck furthest away. Additionally, the full training set, seen in Figure 8.13e, fails to identify the truck located in the right corner of the sample.

It can be observed that all methods face different challenges in identifying the trailers. None of the methods can identify the trailer furthest away, while the Random Sampling method, shown in Figure 8.13b, and the VisLED-Querying Closed-World Mining method, shown in Figure 8.13c, correctly identify one trailer each; however, the Random Sampling method has an incorrectly oriented bounding box. The VisLED-Querying Open-World Exploring method, seen in Figure 8.13d, can identify two trailers. Similarly, the full training set can also identify two trailers, but it is very confused, including a third bounding box that is incorrectly placed.

(a) Ground Truth (b) Random (c) Closed-World Mining

(d) Open-World Exploring (e) Full Training set (No AL)

Figure 8.13. A sample with objects including Truck and Trailer.

In the sample observable in Figure 8.14a, a truck, traffic cones, and temporary traffic barriers are visible under rainy conditions. The rain makes it challenging for the methods to accurately identify the temporary traffic barriers and traffic cones, as the barriers especially almost blend into the grey background of the sample. Among all the methods, only the Random Sampling method, shown in Figure 8.14b, is capable of detecting a few barriers. All the traffic cones, being small and far from the ego vehicle, are not accurately identified by any method, however the labeled once seen in the ground truth in Figure 8.14 are accurately labeled by all methods. The truck is also correctly identified by all methods.

(a) Ground Truth (b) Random (c) Closed-World Mining

(d) Open-World Exploring (e) Full Training set (No AL)

Figure 8.14. A sample with objects including Truck, Barrier and Traffic Cone.

In Figure 8.15a, a sample can be observed containing a car and temporary traffic barriers under dark conditions, with a wet road resulting in bright reflections. It can be seen that all methods accurately identify the car and the barriers in the center of the image. However, the accuracy of the placement of the bounding boxes for the barriers varies among the methods. Additionally, the barrier on the right side of the image and the traffic cone are

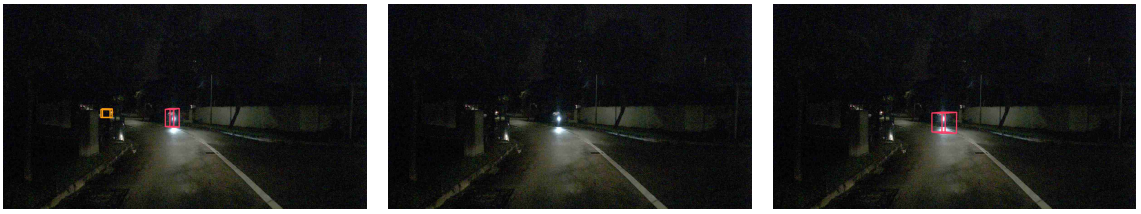
not identified by any of the methods, most likely because of the lack of light in that area of the sample.

(a) Ground Truth (b) Random (c) Closed-World Mining

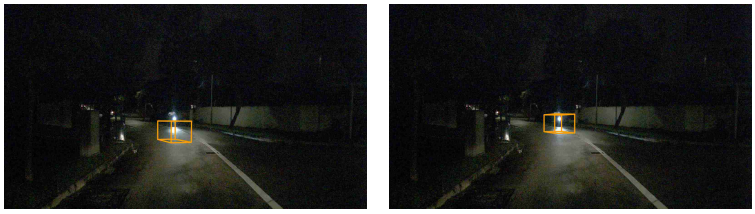
(d) Open-World Exploring (e) Full Training set (No AL)

Figure 8.15. A sample with objects including Car, Barrier and Traffic Cone.

In the sample seen in Figure 8.16a, car and motorcycle objects are visible in a nighttime environment. It can be observed that none of the methods successfully capture the car object. In Figure 8.16b, it is evident that the Random Sampling method fails to identify the motorcycle as well. Conversely, Figure 8.16c shows that the motorcycle is correctly identified by the VisLED-Querying Closed-World Mining method. However, the VisLED-Querying Open-World Exploring method and the full training set, seen in Figures 8.16d and 8.16e respectively, identify an object but wrongly classify it as a car instead of a motorcycle.



(a) Ground Truth (b) Random (c) Closed-World Mining

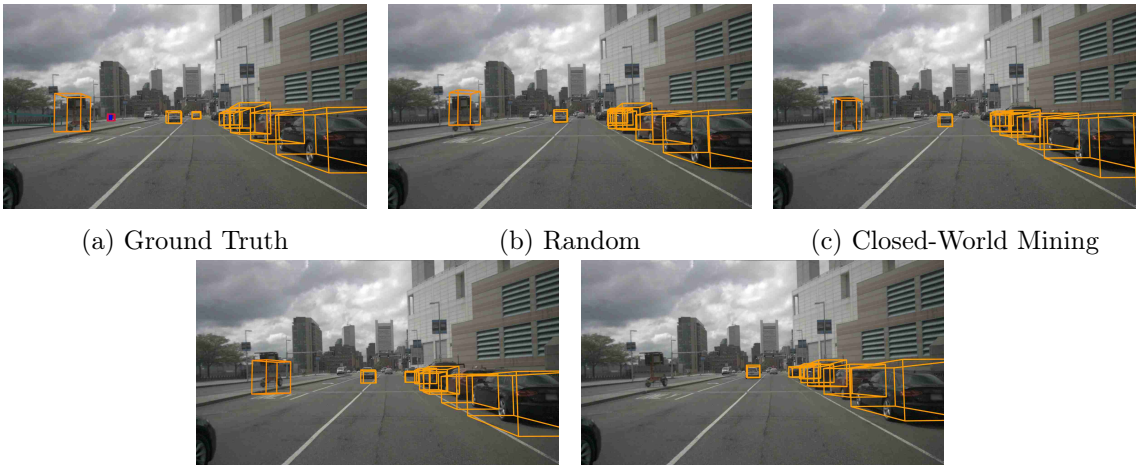


(d) Open-World Exploring (e) Full Training set (No AL)

Figure 8.16. A sample with objects including Car, and Motorcycle.

In the sample seen in Figure 8.17a, objects such as a bicycle, pedestrians, cars, and a trailer are annotated under cloudy weather conditions. It is possible to observe that all algorithms can correctly identify the cars, while they fail to identify the bicycle and pedestrians in the distance.

An interesting observation is that the full training set, seen in Figure 8.17e, cannot identify the trailer, whereas all the AL methods can. Additionally, it is evident that the VisLED-Querying Closed-World Mining method is the most precise, as its bounding box fully encompasses the trailer object. This level of precision is not achieved by the Random Sampling and VisLED-Querying Open-World Exploring methods, seen in Figures 8.17b and 8.17d, respectively.



(a) Ground Truth (b) Random (c) Closed-World Mining

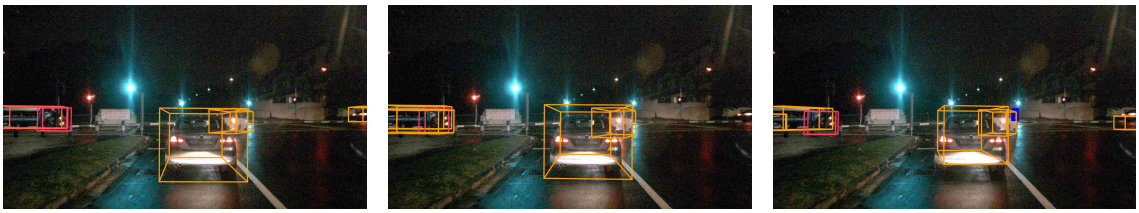
(d) Open-World Exploring (e) Full Training set (No AL)

Figure 8.17. A sample with objects including Car, Pedestrian, Bicycle and Trailer.

In the sample illustrated in Figure 8.18a, the objects include a bus, a motorcycle, and cars in the dark. It is possible to observe that the two closest cars and the motorcycle are correctly labeled by all methods. However, the car that is furthest away is only accurately identified by the full training set and the VisLED-Querying Closed-World Mining method, seen in Figures 8.18e and 8.18c, respectively.

It can be observed that all the methods identify the bus. However, only the VisLED-Querying Open-World Exploring method correctly identifies it as a bus, whereas the other methods mistakenly classify the bus as a car.

A weakness seen in both VisLED-Querying methods for this sample is the presence of false positives for the pedestrian and car classes. Despite this, the algorithms are the only ones to correctly identify the more challenging objects, indicating that the methods are still valuable.



(a) Ground Truth (b) Random (c) Closed-World Mining

(d) Open-World Exploring (e) Full Training set (No AL)

Figure 8.18. A sample with objects including Car, Motorcycle and Bus.

Discussion 9

The results of this study clearly demonstrate the advantages of the VisLED-Querying method over Random Sampling, particularly in terms of data diversity and accuracy. Despite the fact that VisLED-Querying also samples more frequently occurring classes sometimes to the same level as Random Sampling or more, the data remains more diverse. This diversity is crucial as it provides a broader representation of the environment, leading to a more robust model that performs better across various scenarios. The higher performance metrics observed with VisLED-Querying can be attributed to this increased diversity, which enables the model to learn from a wider range of instances and thus generalize better.

Several improvements could be made to the Closed-World Mining (CWM) method to enhance its effectiveness. One significant area of improvement involves allowing one sample to belong to multiple classes. Typically, multiple objects are present in a single sample, and accommodating this would provide a more nuanced understanding of each sample, leading to more accurate class representation. This approach would not only enable the accurate classification of the most prominent objects but also allow for the identification and categorization of partially obscured objects within the sample, potentially leading to a more diverse and comprehensive training set. This adjustment would also resolve issues that arise when including classes like Car and Pedestrian since there are so many objects present from these classes that they would dominate the class distribution.

In further development, it may be beneficial to modify the CLIP model to ensure that the accuracy of one object is not dependent on the accuracy of others within the same sample, as is currently the case. This independence would enhance the model's ability to accurately identify and classify each object individually, without interference from surrounding objects. Alternatively, retraining the CLIP model on data similar to the data seen in the nuScenes dataset, could help specialize it for this specific context. By doing so, the likelihood of false positives in classes such as Temporary Traffic Barrier, Bus, and Bicycle would be significantly reduced.

Exploring open-world practices could further enhance the effectiveness of the VisLED-Querying method. One potential improvement is to set higher thresholds to filter out the most common objects, even if they are rare within their class. For instance, a unique-looking car might be better placed in a cluster rather than in the unique folder if the threshold is appropriately set. This adjustment would ensure that even within rare classes, the most representative samples are selected, thereby improving the model's ability to learn from diverse instances.

Another challenge during this project was the significant impact of time constraints, which limited the number of tests that could be conducted to determine the optimal hyper-

parameters for the model. As a result, the focus was primarily on identifying suitable parameters for the algorithm, which may have influenced the overall performance. Future research should allocate more time for extensive testing to fine-tune the hyper-parameters, thereby enhancing the model's accuracy and efficiency.

One consideration not explored in this thesis is the potential benefit of using an even smaller subset of the nuScenes dataset, as this would reduce training time by decreasing the amount of data processed in each iteration. However, studies have shown that techniques like entropy querying become less effective with smaller subsets, and random sampling tends to perform better when using a subset of up to 4800 samples from the nuScenes dataset [46]. My previous research indicated that when using up to half of the nuScenes dataset with the BEVFusion model, entropy querying outperforms both random sampling and the full training set. Conversely, for the smaller TUMTraf-I dataset, random sampling outperforms entropy querying for the LiDAR model [23]. Given this pattern with both a smaller nuScenes subset and another smaller dataset, the decision was made to use a larger portion of the dataset to achieve better results. Prioritizing a higher possibility of good performance over balancing dataset size and resource utilization, as it was deemed more important.

Conclusion 10

In this thesis, the efficacy of the VisLED-Querying method was explored and compared against Random Sampling for 3D object detection using the nuScenes dataset. The research aimed to enhance the accuracy and diversity of the training data while minimizing the labeled data required. The results demonstrated that VisLED-Querying outperforms Random Sampling, providing higher accuracies and more diverse data representations across various scenarios.

VisLED-Querying, in both the Open-World Exploring and Closed-World Mining scenarios, consistently achieved higher accuracies compared to Random Sampling. This improvement was evident in both the mean average precision (mAP) and nuScenes detection score (NDS), where it outperformed Random Sampling across different training set sizes.

The study demonstrated that using VisLED-Querying achieves high performance with significantly reduced data. At 50 % of the data pool, VisLED-Querying methods reached performance levels close to those obtained with 100 % of the data, thus highlighting the efficiency of active learning approaches in reducing labeling costs while maintaining high model performance. This is primarily due to the diversity of the sampled data, which allows the model to learn from unique instances rather than being overwhelmed by repetitive examples, which can sometimes lead to confusion. Additionally, some individual classes even surpassed the results of the 100 % training set classes. Closed-World Mining achieved this for the Truck class, while the Open-World Exploring method achieved a higher accuracy for the Truck, Traffic Cone, and Construction Vehicle classes.

This success was possible, despite the data distributions of the three methods being quite similar. This is because, even though the classes in the VisLED-Querying methods and Random Sampling are nearly equally represented in many cases, the data sampled for VisLED-Querying is still more diverse, resulting in greater benefits during the training phase.

In conclusion, this thesis has demonstrated that VisLED-Querying is a potent method for active learning in 3D object detection. By leveraging this approach, it is possible to create more accurate and diverse training datasets, ultimately leading to improved model performance while possibly reducing the cost of creating the datasets. The insights gained from this research pave the way for future advancements in active learning strategies and their application in autonomous vehicle systems and other fields requiring robust object detection capabilities.

Bibliography

[1] SAE International. Taxonomy and definitions for terms related to driving autonomation systems for on-road motor vehicles (j3016_202104). https://www.sae.org/standards/content/j3016_202104/, 2014.

[2] The Autonomous Transport Program. Automated vehicles integration into traffic. chrome-extension://bdfcnmeidppjeaggnmidamkiddifkdib/viewer.html?file=https://cdn.gihub.org/umbraco/media/3602/48-autonomous-vehicles.pdf, 2020.

[3] Francisca Rosique, Pedro J. Navarro, Carlos Fernández, and Antonio Padilla. A systematic review of perception system and simulators for autonomous vehicles research. https://www.mdpi.com/1424-8220/19/3/648, 2019.

[4] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022.

[5] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers, 2022.

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *CoRR*, abs/1903.11027, 2019.

[7] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. chrome-extension://bdfcnmeidppjeaggnmidamkiddifkdib/viewer.html?file=https://rgbd.cs.princeton.edu/paper.pdf, 2015.

[8] Hanno Gottschalk, Matthias Rottmann, and Maida Saltagic. Does redundancy in ai perception systems help to test for super-human automated driving performance? *arXiv preprint arXiv:2112.04758*, 2021.

[9] Eduardo Mosqueira-Rey, Elena Hernández-Pereira1, David Alonso-Ríos1, José Bobes-Bascarán1, and Ángel Fernández-Leal. Human-in-the-loop machine learning: a state of the art. https://www.researchgate.net/publication/362748321_Human-in-the-loop_machine_learning_a_state_of_the_art, 2022.

[10] Yilin Wang and Jiayi Ye. An overview of 3d object detection. https://arxiv.org/pdf/2010.15614/, 2020.

[11] Junjie Yan, Yingfei Liu, Jianjian Sun, Fan Jia, Shuailin Li, Tiancai Wang, and Xiangyu Zhang. Cross modal transformer: Towards fast and robust 3d object detection. https://arxiv.org/abs/2301.01283, 2023.

[12] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes. https://www.nuscenes.org/nuscenes#data-annotation, 2019.

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[14] Walter Zimmer, Christian Creß, Huu Tung Nguyen, and Alois C Knoll. Tumtraf intersection dataset: All you need for urban 3d camera-lidar roadside perception. In *26th IEEE International Conference on Intelligent Transportation Systems (ITSC 2023)*. IEEE, 2023.

[15] Walter Zimmer, Christian Creß, Huu Tung Nguyen, and Alois C Knoll. tum-traffic-dataset-dev-kit. https://github.com/tum-traffic-dataset/tum-traffic-dataset-dev-kit, 2023. (Accessed on 05/15/2024).

[16] R. Kesten, M. Usman, J. Houstonand T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, , and V. Shet. Lyft level 5 av dataset 2019. https://wandb.ai/av-datasets/av-dataset/reports/The-Woven-Planet-Lyft-Level-5-Dataset--VmlldzoyNjI0MjE0, 2019.

[17] Veronica Radu, Mihai Nan, Mihai Trascau, David T. Iancu, Alexandra S. Ghita, and Adina M. Florea. Car crash detection in videos. https://www.researchgate.net/publication/353490265_Car_crash_detection_in_videos, 2021.

[18] Sayef. Introduction to nuscenes dataset for autonomous driving. https://medium.com/@msayef/introduction-to-nuscenes-dataset-for-autonomous-driving-2feb7a7e6957, 2023.

[19] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *CoRR*, abs/1912.04838, 2019.

[20] Waymo Open Dataset. Perception. https://waymo.com/open/data/perception/.

[21] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. https://arxiv.org/abs/1703.02910, 2017.

[22] Laboratory for intelligent safe automobiles. https://cvrr.ucsd.edu/team-members.

[23] Ahmed Ghita, Bjørk Antoniussen, Walter Zimmer, Ross Greer, Christian Creß, Andreas Møgelmose, Mohan M Trivedi, and Alois C Knoll. Activeanno3d–an active learning framework for multi-modal 3d object detection. https://arxiv.org/pdf/2402.03235.pdf, 2024.

[24] Ross Greer, Bjørk Antoniussen, Mathias V Andersen, Andreas Møgelmose, and Mohan M Trivedi. The why, when, and how to use active learning in large-data-driven 3d object detection for safe autonomous driving: An empirical exploration. https://arxiv.org/pdf/2401.16634.pdf, 2024.

[25] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. https://arxiv.org/abs/1708.00489, 2018.

[26] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. https://arxiv.org/abs/1906.03671, 2020.

[27] Guohao Li. A principled approach to aggregations. https://medium.com/@pytorch_geometric/a-principled-approach-to-aggregations-983c086b10b3, 2022. (Accessed on 05/16/2024).

[28] Daniel Dworak, Mateusz Komorkiewicz, Paweł Skruch, and Jerzy Baranowski. Cross-domain spatial matching for camera and radar sensor data fusion in autonomous vehicle perception system. https://arxiv.org/abs/2404.16548, 2024.

[29] Sheetal Prasanna and Mohamed El-Sharkawy. Improving mean average precision (map) of camera and radar fusion network for object detection using radar augmentation. In *Proceedings of Seventh International Congress on Information and Communication Technology*, pages 51–61. Springer, Singapore, 2023.

[30] Asam openlabel standard. https://www.asam.net/standards/detail/openlabel/, 2021.

[31] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes annotator instructions. https://github.com/nutonomy/nuscenes-devkit/blob/master/docs/instructions_nuscenes.md#police-officer, 2020.

[32] C. E. Shannon. A mathematical theory of communication. In *The Bell System Technical Journal*, pages 379–423, 1948.

[33] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. Springer, Singapore, 2012.

[34] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. In *Neural Computation 9*, pages 1735–1780, 1997.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. https://arxiv.org/abs/1706.03762, 2017.

[36] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher

Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. https://arxiv.org/abs/2005.14165v4, 2020.

[37] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shane Shixiang Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, and et al. Gpt-4 technical report. https://arxiv.org/abs/2303.08774v4, 2023.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[39] Fatih Karabiber. Hierarchical clustering. https://www.learndatasci.com/glossary/hierarchical-clustering/.

[40] Himanshu Sharma. Hierarchical clustering. https://harshsharma1091996.medium.com/hierarchical-clustering-996745fe656b, 2021.

[41] Ross Greer and Mohan Trivedi. Towards explainable, safe autonomous driving with language embeddings for novelty identification and active learning: Framework and experimental analysis with real-world data sets. *arXiv preprint arXiv:2402.07320*, 2024.

[42] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nutonomy nuscenes devkit. https://github.com/nutonomy/nuscenes-devkit, 2018.

[43] Github - mit-han-lab/bevfusion: [icra'23] bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. https://github.com/mit-han-lab/bevfusion. (Accessed on 05/05/2024).

[44] Geforce rtx 4090. https://www.nvidia.com/en-eu/geforce/graphics-cards/40-series/rtx-4090/. (Accessed on 05/05/2024).

[45] Weights Biases. The ai developer platform. https://wandb.ai/site.

[46] Zhihao Liang, Xun Xu, Shengheng Deng, Lile Cai, Tao Jiang, and Kui Jia. Exploring diversity-based active learning for 3d object detection in autonomous driving. https://arxiv.org/abs/2205.07708, 2022.

Bar Plots A

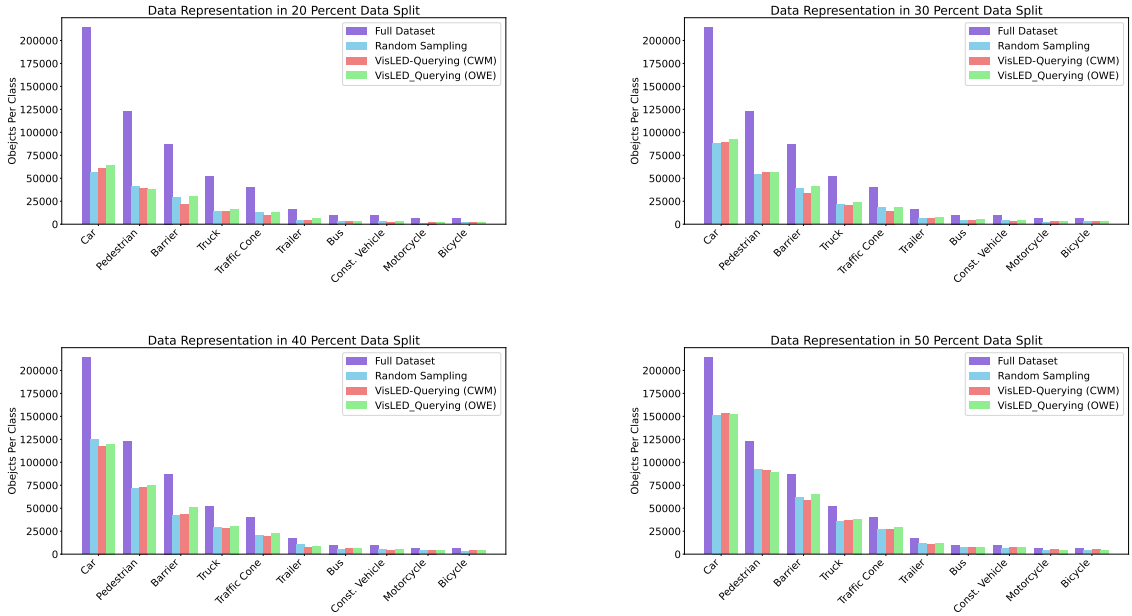


Figure A.1. Class distribution between Random Sampling, VisLED-Querying Open-World Exploring and Closed-World Mining and compared with the full training set.

CVPR Paper Accepted Based on Initial Results

B

This paper was accepted to the Vision and Language for Autonomous Driving and Robotics (VLADR) workshop at the Conference on Computer Vision and Pattern Recognition (CVPR) 2024. The submission is founded on the results obtained from the initial full run of the VisLED-Querying Closed-World Mining method.

Language-Driven Active Learning for Diverse Open-Set 3D Object Detection

Ross Greer * regreer@ucsd.edu
Bjørk Antoniussen * banton19@student.aau.dk
Andreas Møgelmose anmo@create.aau.dk
Mohan Trivedi mtrivedi@ucsd.edu

Abstract

Object detection is crucial for ensuring safe autonomous driving. However, data-driven approaches face challenges when encountering minority or novel objects in the 3D driving scene. In this paper, we propose VisLED, a language-driven active learning framework for diverse open-set 3D Object Detection. Our method leverages active learning techniques to query diverse and informative data samples from an unlabeled pool, enhancing the model's ability to detect underrepresented or novel objects. Specifically, we introduce the Vision-Language Embedding Diversity Querying (VisLED-Querying) algorithm, which operates in both open-world exploring and closed-world mining settings. In open-world exploring, VisLED-Querying selects data points most novel relative to existing data, while in closed-world mining, it mines new instances of known classes. We evaluate our approach on the nuScenes dataset and demonstrate its effectiveness compared to random sampling and entropy-querying methods. Our results show that VisLED-Querying consistently outperforms random sampling and offers competitive performance compared to entropy-querying despite the latter's model-optimality, highlighting the potential of VisLED for improving object detection in autonomous driving scenarios. We make our code publicly available at https://github.com/Bjork-crypto/VisLED-Querying

1. Introduction

Object detection is critical for safe autonomous driving. Data-driven approaches currently provide the best performance in detecting and localizing objects in the 3D driving scene. Detection models perform best on objects which are most represented in driving datasets. This creates challenges when some objects are less represented (minority classes), or unrepresented within the annotation scheme ("novel" objects [1], relevant for "open-set" learning [2]), and becomes especially important when minority objects are most salient to driving decisions [3–6]. Further, from a pragmatic standpoint, the collection, curation, and annotation of such datasets can be extremely expensive [7, 8], motivating the use of heuristics and algorithms which limit annotation efforts while maximizing model learning.

2. Related Research

Active learning methods are driven by a query function which selects relevant data from an unlabeled pool to be annotated and joined to the training set. These methods broadly divide into two classes: uncertainty-based and diversity-based methods [9]. In uncertainty-based methods, data is selected by the query function's assessment of how confusing the datum is *to the existing model*. On the other hand, in diversity-based methods, data is selected by being distinct from existing training data by some measure, and this can be done without consideration of the learning model.

2.1. The Role of Uncertainty and Diversity-Based Methods in Closed and Open Set Learning

In closed-set learning, it is assumed that a system should classify or learn about a fixed set of target classes. By contrast, in open-set learning, the system assumes that it may encounter novel data which belongs to a class unrepresented by its current target set. Naturally, this brings up many research challenges in recognizing this novelty when it appears, determining when to define a new set construct, and integrating new constructs into the learning mechanism.

Here, we suggest that diversity-based methods are particularly well-suited for these open-set learning tasks. Because uncertainty-based methods select relative to their existing world model, there is an inductive bias imposed in relating new data to existing patterns. On the other hand, in diversity-based methods, data is compared only to other data, analogous to unsupervised learning. This does create a tradeoff: closed-set learning excels under uncertainty-driven sampling, since these methods are optimized for the current world model and target set, but cannot treat the world as "open" as diversity-driven sampling. But, critically, we show in this research that diversity-based active learning still provides a benefit to the learning system (even

*Authors contributed equally. R. Greer, B. Antoniussen, and M. Trivedi are with the Laboratory for Intelligent & Safe Automobiles (LISA) at University of California San Diego. A. Møgelmose is with Aalborg University.

if not "optimal" to the particular model and set definition), *and* is suitable for open-set data selection.

2.2. Learning from Vision-Language Representations

Prior research has shown that vision-language representations such as embeddings from contrastive language-image pretraining (CLIP) [10] can be used to identify novelty of an image relative to a set (and, as a bonus, can be decoded into a verbal explanation of novelty) [11]. In our research, we utilize this representation and corresponding ability to select novel images as a proxy for the amount of useful, previously-unexplored information within a complete multimodal driving scene, allowing for an active learning query to select diverse samples based on vision-language encodings of scene images.

3. Algorithm

Here, we present our algorithm named Vision-Language Embedding Diversity Querying (VisLED-Querying), which can be viewed in Figure 1. The algorithm can be used in two different settings:

1. Open-World Exploring: this method imposes no particular class expectations on the data. It is suitable for cases when the model seeks to include information which is most novel relative to data it has seen previously.

2. Closed-World Mining: this method utilizes a zero-shot learning [10] step to sort data between a fixed set of classes before evaluating for novelty, filtering any points estimated to not belong to one of the closed-set classes. This method is suitable for mining new and different instances of existing classes, but may also filter out the most difficult or unusual instances even from known classes if the zero-shot method fails to recognize the object.

Algorithm 1: Open-World Exploring VisLED-Querying

Input: Unlabeled pool of egocentric driving scene images
Output: Updated training set
Embed each egocentric driving scene image from the unlabeled pool using CLIP;
Use hierarchical clustering to separate the embeddings;
Sample new data points from the unclustered set for addition to the training set;

When employing CLIP's [12] zero-shot learning technique for classification, the algorithm examines each sample image to identify objects, that are most likely to belong to predefined classes. As a result, each sample is assigned to a single class, as the zero-shot learning method predominantly identifies one class with high accuracy. In instances where other classes may also be identified, their confidence scores are typically low enough to risk false positives, rendering them inadequate for threshold-based classification. Therefore, a single-class assignment is favored for simplicity and accuracy.

Once the samples for each class have been identified, embeddings will be generated separately for each class, followed by hierarchical clustering. Subsequently, a number of samples will be selected from each class, with a focus on sampling from clusters with minimal data representation. Initially, the algorithm will prioritize unique samples (clusters with only one sample present), matching them with corresponding scene names until the desired number of unique scenes is achieved in the training set. Upon inclusion of all scene-names from unique samples, the algorithm will proceed to clusters containing pairs of images, and so on, until the required number of scenes have been sampled for the training set.

Algorithm 2: Closed-World Mining VisLED-Querying

Input: Unlabeled pool of egocentric driving scene images
Output: Updated training set
Embed each egocentric driving scene image from the unlabeled pool using CLIP;
Encode each class label using a text encoding;
Applying zero-shot learning by maximizing the product of the embeddings, sort the embedded images by class;
For each class, apply hierarchical clustering;
Sample new data points from the unclustered set associated with the desired class, and add to the training set;

4. Experimental Evaluation

4.1. Dataset

We use the nuScenes object detection dataset [13] for our experiments. nuScenes contains 1.4M camera images and 400k LIDAR sweeps of driving data, originally labeled by expert annotators from an annotation partner. 1.4M objects are labeled with a 3D bounding box, semantic category (among 23 classes), and additional attributes. NuScenes comprises 1000 scenes. In order to maintain complete con-

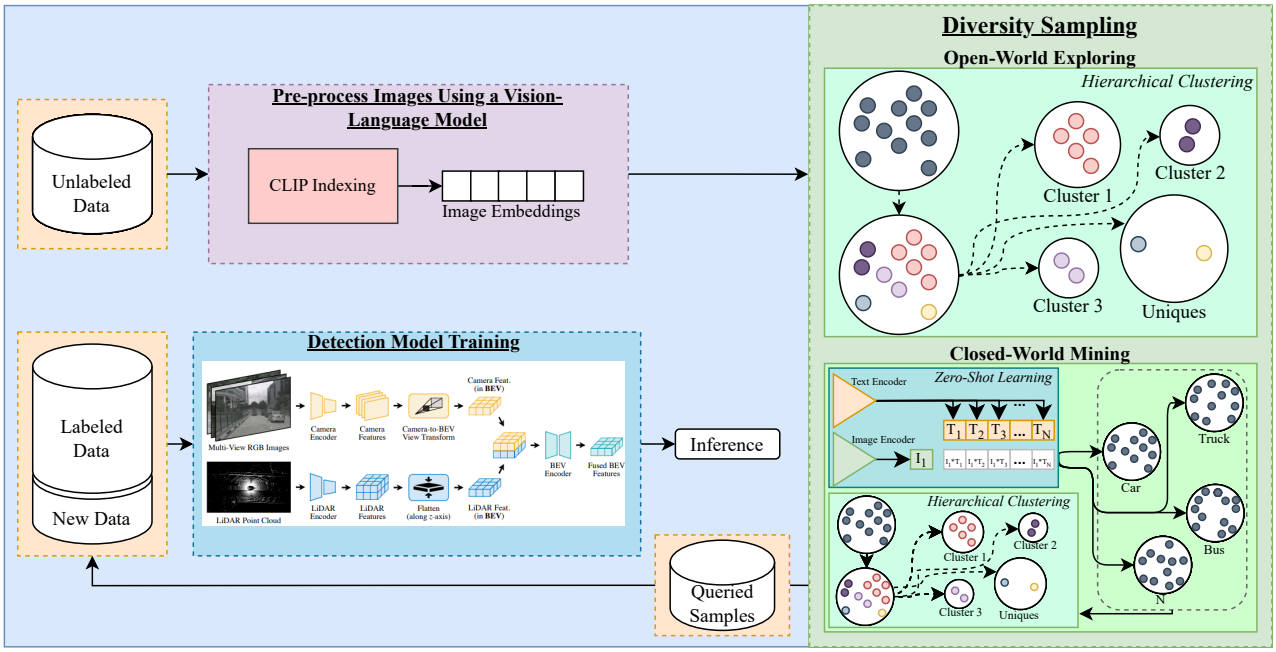


Figure 1. VisLED System Overview. For both Open-World Exploring and Closed-World Mining, the system begins with the processing of the unlabeled data pool into vision-language embedding representations. In Open-World Exploring, these embeddings are clustered and used as the basis for a query. In Closed-World Mining, the embeddings are first used in zero-shot learning to classify scenes based on object appearance, and then further clustered per-class, offering a chance to sample from particular classes which are known to be minority in the labeled training set.

trol over the scenes within the dataset, we modify the fundamental database setup slightly, using the method introduced in [14, 15] to accommodate active learning queries. We use the *trainval* split of the dataset for public reproducibility.

4.2. 3D Object Detection Model

We explore the BEVFusion approach to 3D object detection [16], which has demonstrated notable performance, ranking third in the NuScenes tracking challenge and seventh in the detection challenge. While various methods exist to integrate image and LiDAR data into a unified representation, LiDAR-to-Camera projection methods often introduce geometric distortions, and Camera-to-LiDAR projections face challenges in semantic-orientation tasks. BEVFusion aims to address these issues by creating a unified representation that preserves both geometric structure and semantic density.

In our implementation, we utilize the Swin-Transformer [17] as the image backbone and VoxelNet [18] as the LiDAR backbone. To generate bird's-eye-view (BEV) features for images, we employ a Feature Pyramid Network (FPN) [19] to fuse multi-scale camera features, resulting in a feature map one-eighth of the original size. Subsequently, images are down-sampled to 256x704 pixels, and LiDAR point clouds are voxelized to 0.075 meters to obtain the BEV features necessary for object detection. These modalities are integrated using a convolution-based BEV encoder to mitigate local misalignment between LiDAR-BEV and camera-BEV features, particularly in scenarios of depth estimation uncertainty from the camera mode. For a comprehensive overview of the architecture, including its integration with VisLED-Querying, refer to Figure 1.

4.3. Experiments

We train the BEVFusion model in increasing training set sizes, using three different acquisition modes: (1) Random Sampling, (2) Entropy-Querying, and (3) VisLED-Querying with Closed-Set Mining setting. As expected, active learning strategies outperform the random baseline, and the entropy-querying method is dominant due to its nature of optimizing uncertainty with respect to the model, as opposed to VisLED's model-agnostic sampling. Yet, as illustrated in Table 1, VisLED still stays consistently ahead of random sampling, and offers a 1% gain over random sampling mAP at 50% of the data pool, all without requiring *any* model training or inference.

5. Discussion and Conclusion

Our presented learning method, VisLED-Querying, samples without any information about the model. This enables

Labeled Pool		mAP			NDS		
Rounds	%	Random	Entropy	VisLED	Random	Entropy	VisLED
1	10%	30.95	31.06 (+1.06)	29.14 (-1.81)	33.53	34.09 (+0.56)	32.16 (-1.37)
2	20%	38.00	40.41 (+2.41)	40.76 (+2.76)	40.14	41.85 (+1.71)	41.18 (+1.04)
3	30%	44.94	45.57 (+0.63)	45.01 (+0.07)	48.41	50.11 (+1.7)	49.40 (+0.99)
4	40%	47.73	49.24 (+1.51)	49.21 (+1.48)	53.10	53.80 (+0.7)	53.64 (+0.54)
5	50%	**49.90**	**63.88** (+13.98)	**51.05** (+1.15)	**55.64**	**64.85** (+9.21)	**56.45** (+0.81)
	100%	**52.88**			**58.73**		

Table 1. This table shows the mean average precision (mAP) and nuScenes detection score (NDS) metrics for the random sampling, entropy-querying, and VisLED-querying (Closed-World Mining) in every round. It also shows the mAP and NDS scores for the full training split when trained using one GPU. Both the entropy-querying and VisLED methods outperform random sampling consistently, and reach nearly the same level of performance as 100% of the data at just the 50% data point, showing faster learning than the baseline method.

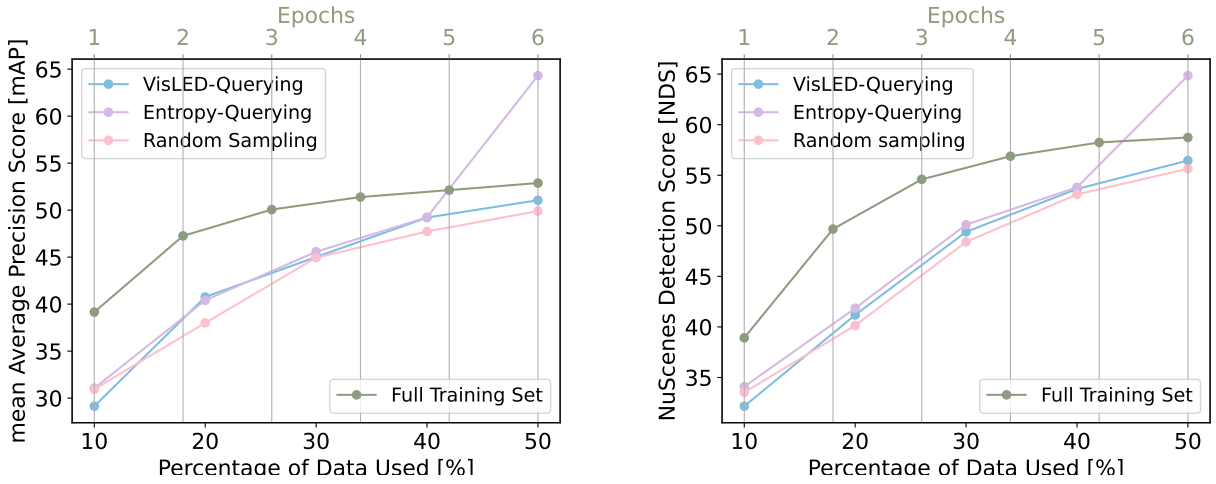


Figure 2. Performance of BEVFusion in 3D Object Detection on nuScenes at different training set sizes, using three different learning strategies. Simultaneously, we chart the learning of BEVFusion on the full training set, over the course of six epochs (top horizontal axis) to give an impression of the asymptotic performance limit that may be expected of the model. We observe that the active learning methods move towards this asymptote sooner than random sampling, and that VisLED maintains a margin over random sampling throughout.

VisLED to select novel, informative data points, to the extent that novelty is visibly identifiable, for *any* model. The benefit this offers is that a data point may need to be annotated only once, and can then be used in a variety of models for additional autonomous driving tasks instead of sampling and possibly forming an entirely different set for annotation. While these gains may be marginal in the current data setting ($<$ 1000 scenes), at scale, these performance gains may translate to serious reductions in annotation costs and safety-critical detection failures. Further, VisLED offers one key possibility that is otherwise limited on uncertainty-driven approaches: VisLED will recommend unique samples without any prior assumptions on class taxonomy, making it especially suited to open-set learning, where new classes may be introduced at any time. This capability, when paired with methods of self- or semi-supervised learning for object detection by fusing LiDAR and camera [20], may prove especially beneficial in identifying and learning from novel encounters. In future research, we plan to experiment on the effectiveness of VisLED in multi-task learning settings [21], experiments on other benchmark datasets [22], and experiments in open-set and continual learning.

References

[1] Valerie Chen, Man-Ki Yoon, and Zhong Shao. Task-aware novelty detection for visual-based deep learning in autonomous systems. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11060–11066. IEEE, 2020. 1

[2] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open

set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772, 2012. 1

[3] Ross Greer, Jason Isa, Nachiket Deo, Akshay Rangesh, and Mohan M Trivedi. On salience-sensitive sign classification in autonomous vehicle path planning: Experimental explorations with a novel dataset. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 636–644, 2022. 1

[4] Eshed Ohn-Bar and Mohan M Trivedi. What makes an on-road object important? In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3392–3397. IEEE, 2016. 1

[5] Ross Greer, Akshay Gopalkrishnan, Nachiket Deo, Akshay Rangesh, and Mohan Trivedi. Salient sign detection in safe autonomous driving: Ai which reasons over full visual context. In *27th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration*, number 23-0333, 2023. 1

[6] Ross Greer, Akshay Gopalkrishnan, Jacob Landgren, Lulua Rakla, Anish Gopalan, and Mohan Trivedi. Robust traffic light detection using salience-sensitive loss: Computational framework and evaluations. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–7. IEEE, 2023. 1

[7] Aseem Behl, Kashyap Chitta, Aditya Prakash, Eshed Ohn-Bar, and Andreas Geiger. Label efficient visual abstractions for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2338–2345. IEEE, 2020. 1

[8] N Kulkarni, A Rangesh, J Buck, J Feltracco, M Trivedi, N Deo, R Greer, S Sarraf, and S Sathyanarayana. Create a large-scale video driving dataset with detailed attributes using amazon sagemaker ground truth. 2021. 1

[9] Sanjoy Dasgupta. Two faces of active learning. *Theoretical computer science*, 412(19):1767–1781, 2011. 1

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[11] Ross Greer and Mohan Trivedi. Towards explainable, safe autonomous driving with language embeddings for novelty identification and active learning: Framework and experimental analysis with real-world data sets. *arXiv preprint arXiv:2402.07320*, 2024. 2

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, 2021. 2

[13] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2

[14] Ahmed Ghita, Bjørk Antoniussen, Walter Zimmer, Ross Greer, Christian Creß, Andreas Møgelmose, Mohan M Trivedi, and Alois C Knoll. Activeanno3d–an active learning framework for multi-modal 3d object detection. *arXiv preprint arXiv:2402.03235*, 2024. 3

[15] Ross Greer, Bjørk Antoniussen, Mathias V Andersen, Andreas Møgelmose, and Mohan M Trivedi. The why, when, and how to use active learning in large-data-driven 3d object detection for safe autonomous driving: An empirical exploration. *arXiv preprint arXiv:2401.16634*, 2024. 3

[16] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 2774–2781. IEEE, 2023. 3

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, , and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted window. *ICCV*, 2021. 3

[18] Yan Yan, Yuxing Mao, , and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018. 3

[19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detectio. *CVPR*, 2017. 3

[20] Aral Hekimoglu, Michael Schmidt, and Alvaro Marcos-Ramiro. Monocular 3d object detection with lidar guided semi supervised active learning. In *Proceedings of the IEEE/CVF Winter Conference on*

Applications of Computer Vision, pages 2346–2355, 2024. 4

[21] Aral Hekimoglu, Philipp Friedrich, Walter Zimmer, Michael Schmidt, Alvaro Marcos-Ramiro, and Alois Knoll. Multi-task consistency for active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3415–3424, 2023. 4

[22] Walter Zimmer, Christian Creß, Huu Tung Nguyen, and Alois C Knoll. Tumtraf intersection dataset: All you need for urban 3d camera-lidar roadside perception. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1030–1037. IEEE, 2023. 4