Seeing Through Occlusion: Innovative Techniques for Reliable LiDAR-based Object Detection

William Guimont-Martin (111 175 409)
September 21, 2023

Supervised by Prof. P. Giguère
and co-supervised by Prof. F. Pomerleau

1 Introduction

In recent years, the advancement of autonomous vehicle technologies has garnered significant attention, promising a revolutionary transformation in transportation systems as well as in applications of mobile robotics. A crucial aspect of enabling safe and reliable autonomous navigation is robust object detection, which relies heavily on accurate perception of the environment. Light Detection and Ranging (LiDAR) has emerged as a cornerstone sensor for such perception due to its ability to capture detailed 3D spatial information. However, one of the persistent challenges in LiDAR-based object detection is occlusion — the obstruction of objects by other objects, itself or the environment [1]. This phenomenon introduces uncertainty and ambiguity, potentially leading to erroneous detection and compromised decision-making for autonomous vehicles. Occlusion will make objects appear incomplete in points clouds, i.e., objects will only be partially visible, thus making them incomplete and harder to detect using 3D object detectors [1].

This Ph.D. project proposal seeks to address the critical challenges posed by occlusion by exploring innovative techniques to enhance object detection accuracy in occluded scenarios. Our first objective is to improve object detection of occluded objects by developing a novel mask-based bird's-eye view (BEV) object detection neural network architecture. This architecture is designed to simultaneously detect and complete objects, thereby improving object recognition in occlusion-prone contexts. The second objective revolves around addressing a gap in existing autonomous vehicle datasets by introducing an innovative dataset that employs multi-LiDAR point clouds and integrates drone imaging to achieve a top-down view (BEV) of the scene. Lastly, the third objective involves exploring the application of multi-modal self-supervised learning within the domain of autonomous vehicles, leveraging the dataset of the second objective. By investigating occlusion and developing novel architectures, datasets and training techniques, this proposal aspires to overcome the many challenges posed by LiDAR point clouds for deep learning.

This proposal is structured as follows: the essential theoretical groundwork is presented in Section 2, while the research question itself is defined in Section 3. Section 4 offers an overview of related work relevant to the proposed research avenues. The three primary objectives, along with the respective methodologies, are elaborated upon in Section 5. A detailed schedule for each

objective is outlined in Section 6. Finally, the proposal concludes in Section 7, encapsulating the comprehensive objectives and potential contributions of the research.

2 Theory

This section reviews the fundamental concepts necessary for comprehending the main research question of this proposal, which is detailed in Section 3. A more extensive examination of the relevant literature will be conducted in Section 4.

2.1 Point Clouds and Their Challenges for Deep Neural Networks

Handling point cloud data presents several challenges for deep neural networks, stemming from the inherent characteristics of these data structures. A point cloud is a set of 3D points in a metric space — more precisely for LiDAR data, a set of points in $\mathbb{R}^3$, alongside the Euclidean distance metric. When available, these points might also include supplementary attributes, such as return intensity (i.e., reflectance). However, this kind of data introduces a set of distinct challenges, classified into three primary categories by Bello *et al.* [2]. Point clouds are irregular, unstructured, and unordered.

Irregularity Point clouds are characterized by uneven spatial sampling, leading to differing point densities across various regions of a scene. For instance, areas situated closer to the LiDAR sensor often exhibit a higher density of points compared to those farther away. This phenomenon is particularly pronounced in the context of autonomous vehicle applications, given the complex and dynamic nature of urban landscapes. Point clouds in these scenarios often exhibit sparser distributions of points in space [3]. This non-uniform sampling makes point clouds harder for neural networks to handle, necessitating the capability to manage diverse levels of detail and the efficient capture of spatial information across the entirety of the point cloud.

Unstructured Unlike structured data like images, which follow a regular grid arrangement of evenly spaced pixels, point clouds lack this kind of structure. Points within a point cloud can occupy any 3D position within the sensor's range. This unstructured nature makes it challenging to apply traditional grid-based convolutional operations, as commonly used in computer vision. Neural networks designed for point clouds must account for this lack of structure.

Unorderness Point clouds, being a set of points, inherit the unordered nature of sets. Consequently, there exists no inherent ordering among the points. This lack of order presents a significant challenge when it comes to designing neural network architectures handling point clouds. Operations within such neural networks must be invariant to point permutations, as any permutation of a point cloud still represents the same underlying point cloud.

Overcoming these challenges requires the development of specialized neural network architectures capable of processing such data. Section 4.1 provides an overview of some methods addressing these challenges.

2.2 Occlusion in LiDAR Point Clouds

While the challenges outlined in Section 2.1 are related to the type of data of point clouds as a set of points in $\mathbb{R}^3$, the concept of occlusion pertains to the semantic aspects of the data, i.e., what these points represent. In the context of LiDAR point clouds, occlusion constitutes a significant challenge that can impact the accuracy and reliability of object detection [1]. Occlusion refers to the phenomenon wherein objects are obscured or partially concealed by other objects, the environment, or even themselves. This phenomenon can lead to incomplete representations of objects within the point cloud, rendering their detection more challenging [1], [4]. Occlusion can be classified into different types based on its nature and source [1], each contributing to the complexity of comprehending the environment. In addition to occlusion, signal miss can also result in a partial representation of objects in point clouds.

Self-Occlusion Self-occlusion occurs when one side of an object obstructs the view of its opposite side, leaving only the portion facing the LiDAR sensor visible within the point cloud. This type of occlusion is inherent and affects every object within a LiDAR scan. Given that solely one facet of the object is observable in the LiDAR scan, accurately determining the object's complete shape becomes a challenging endeavour. An example of self-occlusion is illustrated in Figure 1, where there are points in red only on the LiDAR-facing side, leaving its backside empty.

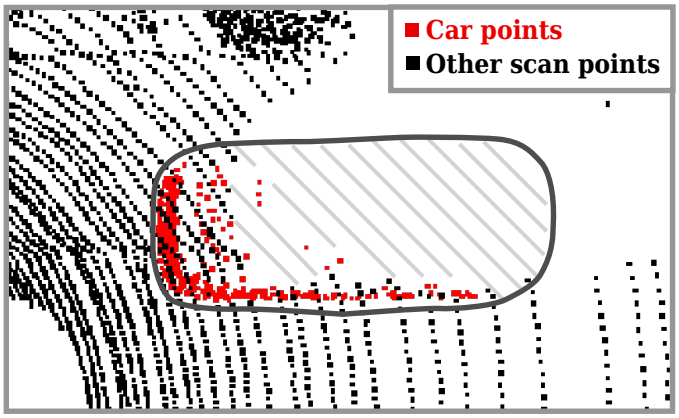


Figure 1: Example of self-occlusion in SemanticKITTI[5] from a BEV perspective. The red points denote a car instance, with its complete footprint outlined in grey. The side of the car facing the LiDAR sensor hides the remainder of the object, resulting in an absence of points on the rest of the object and behind it.

External-Occlusion External-occlusion occurs when one object obstructs the visibility of another object within the scene. The occluding object blocks the LiDAR's beams from reaching the objects, creating frustums of occluded space. For instance, a large car might occlude a smaller vehicle behind it, rendering the hidden object partially or entirely invisible in the point cloud. Additionally, this type of occlusion introduces ambiguity regarding whether empty regions of space in

the point cloud are indeed empty or instead occluded due to the presence of occluding objects. An example of external-occlusion can be seen in Figure 1, where no points are present on the ground behind the vehicle.

Signal Miss In addition to self- and external-occlusion, signal misses can also result in missing points on objects. A signal miss occurs when reflective materials deflect the LiDAR's laser beams away from the sensor or when low reflectance causes the absence of reflected signals. This can be caused by factors such as reflective properties, transparency, or the sensor's viewing angle. Consequently, objects incapable of effectively reflecting LiDAR signals may manifest as gaps or voids within the point cloud data. Signal miss further complicates object detection by contributing to incomplete point cloud representations.

The overarching result of occlusion and signal miss is that they make objects appear incomplete within point clouds, hampering the ability of object detectors to accurately detect them [1], [4].

3 Research Problem

Object detection in point clouds is a crucial task for a wide range of applications, notably in the context of autonomous vehicles. Point clouds are often generated by LiDAR sensors, capturing intricate spatial information about the environment. The objective of object detection is to accurately identify and locate various objects, such as vehicles, pedestrians, and cyclists, within a 3D point cloud. This task involves predicting both the spatial location and the class of detected objects. An illustration of an object detection task is provided in Figure 2. This task is particularly challenging due to the dynamic and complex and dynamic nature of the environments where autonomous systems operate [3].

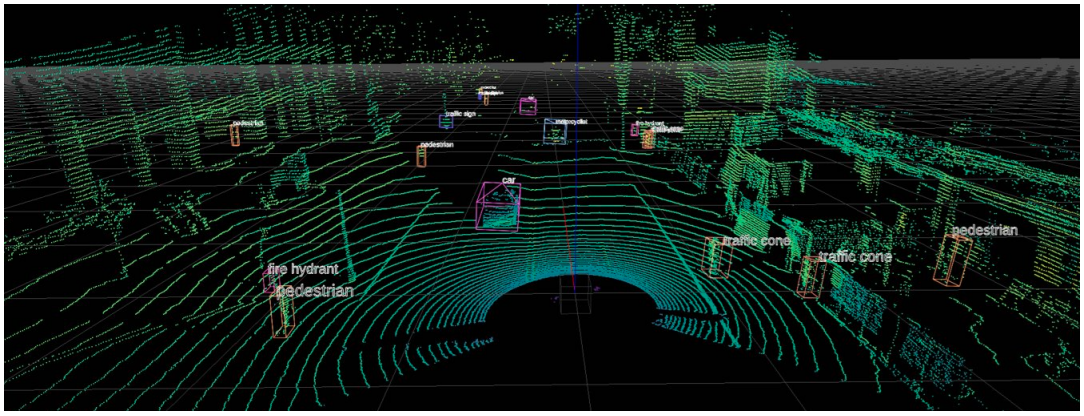


Figure 2: Illustration of an object detection task. The primary goal of object detection is to accurately locate and classify objects (e.g., vehicles, pedestrians, and cyclists) within a 3D point cloud. We show here detected objects with bounding boxes and labels with their respective class names. Image created using data from Déziel *et al.* [6].

Among the many applications of deep learning to point cloud data [2], our research proposal focuses solely on real-time dynamic object detection for autonomous vehicles. The application of these techniques will vary greatly depending on the field they are used in. For instance, point cloud analysis in geomatics and surveying primarily revolves around static structures processed in an offline manner. As such, some techniques are not possible to use either because they only work on static objects (i.e., building maps by registering point clouds), or cannot be used in a real-time setting. The analysis of point clouds also varies greatly depending on the source of the data. Datasets used for object classification and part segmentation, such as ModelNet40 [7] and ShapeNet [8], generally consist of point clouds acquired from scanned objects or 3D meshes. Thus, these datasets provide points from all sides of the objects, which is not usually the case in autonomous vehicle contexts where we are "inside" of the scene and suffer from occlusion. Indoor datasets like the Indoor Stanford 3D Indoor Dataset (S3DIS) [9] and ScanNet [10] offer denser point clouds in simpler environments. They contrast with outdoor point clouds, which are characterized by increased complexity, sparser point clouds and dynamic scenes [3], which exacerbated challenges highlighted in Section 2. Also, LiDAR point clouds are effectively 2.5D representations because

they only capture portions of the underlying shapes [1]. Object detection for autonomous vehicles thus provides unique challenges that require innovative techniques to solve, notably for handling dynamic objects, occlusion and the high complexity of environments.

As discussed in Section 2.2, occlusion poses a significant challenge to deep neural networks operating on point clouds. Objects often appear incomplete, making detection more challenging and exacerbating the inherent complexities in point cloud analysis. These challenges can lead to potential accidents [4], and thus the development of more robust object detection architectures is essential for improved performances and safer autonomous vehicles.

In addition to occlusion challenges, the annotation of 3D data remains a hurdle due to the time-consuming and expensive nature of data labelling [5], [11]. Most autonomous vehicle datasets are collected in North American and European cities, resulting in a need for simpler data annotation processes to facilitate the collection of supplementary data in new environments and scenarios. Wilson *et al.* [12] highlights the importance of mining interesting scenarios in varied environments for autonomous vehicle datasets. This prompts a novel data acquisition technique that deliberately gathers data that is complementary (i.e., in object types, viewpoints, and environments) to larger-scale datasets in a cost-effective manner. This would help to extend the application of point cloud-based object detection to regions or scenarios excluded from major dataset efforts, such as remote or forested areas, and enhance the overall performance of deep learning models.

In light of these challenges, our research proposal outlines three primary research avenues.

Joint Object Detection and Completion We introduce MaskBEV, a novel approach aimed at bridging the gap between point cloud and computer vision architectures. MaskBEV, detailed in Section 5.1, is designed to simultaneously detect and complete objects from a BEV perspective. By training MaskBEV to predict complete BEV instance masks (i.e., binary images representing the complete footprint of each detected object from a BEV), we aim to bolster object detection's resilience to various forms of occlusion.

Multi-LiDAR BEV Dataset Our proposal introduces **B**ird's-**E**ye **V**iew **Occ**lusion (BEVOcc), a novel dataset employing multiple static LiDARs and a drone-mounted BEV camera. The inclusion of multiple LiDAR viewpoints, placed on either side of streets, enables the acquisition of point clouds from diverse perspectives, and thus with reduced occlusion. Moreover, the BEV viewpoint offers valuable insights into the complete scene, streamlining the annotation process. This dataset seeks to complement existing autonomous vehicle datasets by incorporating multiple LiDAR viewpoints in interesting and varied scenarios. BEVOcc is aimed at gaining invaluable insights into occlusion and its impact on deep neural network performance, particularly in object detection tasks. Also, this new dataset will allow us to evaluate the benefits of adding complementary data, acquired and annotated in a lower cost easily reproducible manner manner, to larger datasets. This dataset will also provide a fertile ground for testing training algorithms for vehicle-to-vehicle (V2V) applications, as the usage of multiple LiDAR scans is an invaluable tool to counter the effects of occlusion and the short usable perception range of LiDARs with [4].

Multi-Modal Self-Supervised Learning The application of Self-Supervised Learning (SSL) techniques has witnessed considerable success in various domains of machine learning [12]. SSL techniques aim to extract knowledge from unlabelled data, providing a means to mitigate the challenges associated with annotating 3D data [3]. By capitalizing on the BEV information available in BEVOcc, we intend to train networks to develop a form of "common sense" [13] about the structure of 3D scenes, thereby enhancing object detection's robustness in occluded scenarios. Furthermore, SSL can effectively exploit the richness of the diverse modalities within BEVOcc, presenting a promising avenue for improving object detection. This underscores SSL as a compelling research pathway for enhancing object detection within the autonomous vehicle domain.

In summary, the central research question this proposal aims to address is:

How can we enhance LiDAR-based object detection in occluded scenarios?

The proposed objectives, detailed in Section 5, each present a project that contributes to answering this overarching question. These objectives encompass training networks to complete occluded objects (Section 5.1), introducing a new dataset with multi-LiDAR perspectives and BEV imagery (Section 5.2), and investigating the potential of SSL techniques to bolster object detection in occluded environments (Section 5.3).

4 Related Work

This section provides an overview of the existing research on deep learning techniques applied to point clouds. We will first explore the diverse range of deep learning approaches that have been developed to handle point cloud data in Section 4.1. Following that, we will delve into the realm of 3D object detection models (Section 4.2) and discuss strategies for addressing occlusion challenges in deep neural networks (Section 4.3). Finally, we will review widely used datasets in the domain of autonomous vehicle research, particularly for training and evaluating object detection models, in Section 5.2.

4.1 Deep Learning on Point Clouds

Point cloud data presents unique challenges, as discussed in Section 2, due to its irregular, unstructured, and unordered nature. These characteristics require the development of specialized techniques to enable deep neural networks to effectively handle such data. In this section, we draw inspiration from the classification proposed by Bello *et al.* [2] and Guimont-Martin [14] to present a taxonomy of these techniques. Figure 3 illustrates this taxonomy.

These techniques can be broadly categorized into two main groups: structured and unstructured methods. Structured approaches involve imposing a specific structure onto point clouds to make them more amenable to neural network processing. On the other hand, unstructured approaches focus on designing novel neural network architectures specifically tailored for handling raw point cloud data. The categorization presented in this section draws inspiration Guimont-Martin [14].

4.1.1 Structured Approaches

Structured approaches tackle the inherent lack of organization in point cloud data by imposing a predefined structure onto the raw point cloud data. These structures facilitate the application of deep learning techniques. This can be done by voxelizing the point cloud into a 3D regular grid representing a volume of cube-shaped cells (i.e., voxels), making possible the application of Convolutional Neural Network (CNN). Multi-view approaches instead take on the problem from a computer vision point of view, representing the 3D point cloud as a series of 2D views. Another approach is to project the point cloud onto a regular grid of points (i.e., a lattice), on which a convolution operator can be defined.

Voxel-Based One common structured approach is voxelization, which involves partitioning the volume containing the point cloud into small cuboids known as voxels. Each voxel contains a subset of the input point cloud data. These subsets can be represented as embedding vectors (i.e., features) using various encoding methods. For instance, encoding can be based on binary occupancy [15], point density within each voxel [16], or applying a PointNet per voxel [17]. This representation

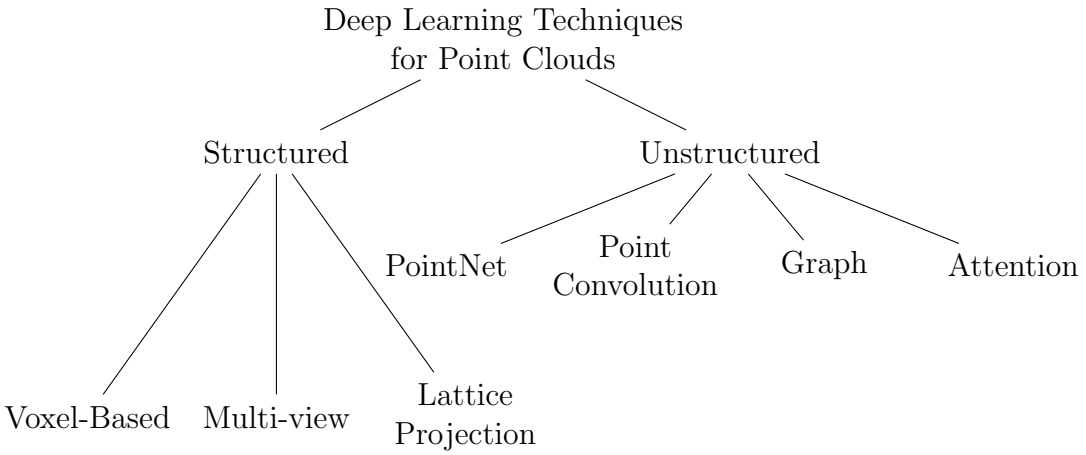


Figure 3: Taxonomy of deep learning techniques for point clouds. This taxonomy classifies techniques into two main groups: structured and unstructured approaches. Structured approaches, such as Voxel-Based, Multi-view, and Lattice Projection, impose a structures to point clouds. In contrast, unstructured approaches, including PointNet, Point Convolution, Graph, and Attention-based methods, are designed to handle raw point cloud data directly. This figure is adapted and translated from Guimont-Martin [14].

effectively transforms the irregular point cloud into a structured, regular 3D grid, enabling the application of 3D convolutions, similar to those used in image processing.

Despite their conceptual simplicity, voxel-based methods have certain drawbacks. They often demand substantial memory resources due to the necessity of storing a large 3D grid with fine resolution to capture details accurately. Additionally, computational costs tend to be higher because convolutions need to be applied across multiple dimensions. LiDAR scans from autonomous vehicles, for example, are frequently sparse [3], leading to numerous empty voxels [18]. To mitigate these issues, techniques such as sparse voxel representations and sparse convolutions [19], [20] can be employed to avoid storing empty voxels and reduce unnecessary computation. However, in any case, if the voxel resolution is too coarse, voxelization-based methods may suffer from quantization artifacts.

Some models using voxelization for object detection for autonomous vehicle applications, like PointPillars [21] and PIXOR [18], use a 2D voxel grid aligned parallel to the ground, treating each voxel as a vertical rectangular pillar. This approach effectively generates a 2D pseudo-image representing the point cloud from a BEV perspective, enabling the application of standard 2D convolutions.

Multi-view Multi-view approaches sidestep the challenges of 3D data by translating the point cloud into a series of 2D views. These views are generated by placing virtual cameras around the point cloud, effectively projecting the 3D data into 2D space. While each individual view lacks depth information, 3D information can be inferred by leveraging relationships between multiple

views. Typically, computer vision models like CNN are applied independently to each 2D view, and the information is fused across views using pooling operations.

Multi-view approaches do not suffer from quantization artifacts and are computationally efficient, as they often rely on CNN. However, they are often limited to object datasets (i.e., 3D scans of objects [7], [8]), which are simpler than the complex outdoor scenes encountered in autonomous driving scenarios [3]. Consequently, multi-view techniques are predominantly used for object classification [22], [23]. Although it is possible to adapt multi-view methods for object detection [24], they are less common in tasks related to autonomous vehicles, which involve more intricate 3D scenes.

Lattice Projection Lattice projection-based methods address the challenge of point clouds' unstructured nature by projecting the raw points onto a higher-dimensional regular grid of points, known as a lattice. This lattice projection provides a structured foundation on which convolution operations can be defined. For example, in the case of point clouds with (x, y, z) coordinates and (r, g, b) colour information, a lattice projection-based network would project each point onto a 6-dimensional grid, encompassing (x, y, z, r, g, b) coordinates. Subsequently, convolutions can be applied within this grid [25]. Other approaches, such as that of Rao *et al.* [26], project points onto a fractalized regular icosahedron lattice, approximating the surface of a sphere. This approach helps maintain rotation invariance. The structure of the underlying lattice can help the model to maintain an invariance to certain transformations.

4.1.2 Unstructured Approaches

While structured approaches adapted the data representation to fit the needs of neural networks, unstructured approaches instead adapted the design of the networks to better fit point clouds. This often involves the use of innovative techniques instead of traditional convolution operations. Such approaches typically hinge on the underlying properties of point clouds. Several notable unstructured approaches include PointNet-based networks, point convolution-based methods, graph neural networks, and attention-based networks.

PointNet Point clouds, fundamentally, are sets of points, making it logical to design networks capable of directly operating on sets. To handle the unordered nature of sets, these networks should be invariant to the permutation of their input points. PointNet [17] achieves this by utilizing per-point Multi-Layer Perceptrons (MLPs) and the max-pooling function. Both of these components are symmetric functions that remain invariant to the order in which points are presented to the network.

In the PointNet architecture, a linear layer is applied individually to each point within the point cloud. Then, max-pooling is used to summarize the entire point cloud into a single vector representation, which can be used for classification or combined with per-point features to do per-point semantic segmentation. To ensure invariance to affine transformations, PointNet employs a

specialized sub-network to predict an affine transformation that can be applied to the point cloud, to transform the point cloud into a canonical pose that makes further processing easier.

Building upon PointNet, PointNet++ [27] introduces a hierarchical structure that leverages local details within the point cloud. This hierarchy enhances the network's ability to capture intricate patterns and relationships within the data.

Point Convolution While structured methods impose a specific structure onto point clouds to enable convolution operations, point convolution takes a different approach by adapting convolution to raw point clouds. Methods like FKAConv [28] perform convolutions on the K nearest neighbours of a given point. This convolution is achieved by introducing a projection operation that maps the space of points to the space of the convolution kernel. This projection aligns points with a point convolution kernel, enabling the application of the convolution operation. Since the size of this neighbourhood can vary based on point density, normalization operations are applied to ensure consistency and stability throughout the convolution process.

Several other approaches, such as KPConv [29], PointConv [30], and PointCNN [31], also generalize convolution to operate on individual points within point clouds.

Graph Graph-based methods treat point clouds as graphs and apply graph convolution operations. For instance, DGCNN [32] constructs a graph by connecting each point to its K nearest neighbours and then applies a Graph Neural Network (GNN) to extract information about local structures within the point cloud. This process generates a new feature graph, where feature vectors become new points, and the operation can be iteratively repeated.

Other approaches also operate on K nearest neighbours graphs, such as KCNet [33], ClusterNet [34], and PointGCN [35]. Some methods, like RG-CNN [36], use complete graphs, but this inherently limits them to object-related tasks involving smaller point clouds.

Attention Recent advances in transformer-based models [37] have positioned transformers as essential tools in various machine learning domains, including computer vision [38], [39]. Transformers excel at processing sets of tokens, making them invariant to the permutation of their inputs.

Taking advantage of this property, Point Transformer [40] applies a transformer-based model directly to point clouds. Similarly, Point-BERT [41] harnesses the affinity of transformer models and adapts the self-supervised training technique from BERT [42] for point cloud data. Other methods, such as 3DETR [43], M3DETR [44], and BEVFusion [45], also incorporate transformers into their architecture to enhance their capabilities in processing point clouds.

4.2 3D object detection

3D object detection in point clouds is a vast domain encompassing a large range of techniques, primarily focusing on the prediction of sets of bounding boxes. These methods can be broadly

categorized into anchor-based and anchor-free detectors. In addition, this section explores mask-based detection, an approach that has recently garnered significant attention in the field. The inspiration for this section comes from the work of Guimont-Martin *et al.* [46], as they provide a comprehensive overview of object detection architectures, with a special focus on the interests of this proposal.

4.2.1 Anchor-based

Anchor-based detection methods rely on predefined box proposals, known as anchors, to predict object location and size. These approaches can be classified into single-stage and two-stage methods. Anchors represent a set of potential object locations and sizes that serve as the foundation for the network's predictions. Consequently, anchors encapsulate substantial prior knowledge (i.e., hyperparameters) about the objects to be detected, including their location, dimensions, position relative to the ground, and viewing angle [21]. It's worth noting that correctly tuning these parameters is crucial, as improper tuning can lead to sub-optimal performance as the predictions are regressed relative to the anchors.

Single-stage methods Single-stage approaches detect objects by directly regressing 3D bounding boxes relative to detection anchors in a single step. This simplicity has led several architectures to embrace the single-stage approach. Notable examples include VoTr-SSD [47], PointPillars [48], 3DSSD [49], SE-SSD [50], and GLENet [51].

Two-stage methods Two-stage architectures trace their origins back to the framework introduced by Faster R-CNN [52]. The process begins with an initial stage that extracts regions of interest (RoIs) proposals using a Region Proposal Network (RPN). Numerous methods have built upon this approach, including Fast Point R-CNN [53], PV-RCNN++ [54], Voxel R-CNN [55], and BtcDet [56] These approaches rely on anchors to define possible RoIs and may encounter similar limitations as single-stage detectors.

4.2.2 Anchor-free

Anchor-free detectors bypass the use of anchors and instead predict keypoint locations for object detection. In the context of 3D object detection, a common approach involves the direct prediction of the centers of detected objects [57]–[62]. Alternatively, 3D objects can also be detected by predicting the corners of their bounding boxes [63] or a few keypoints distributed around the shape of vehicles [64]. Anchor-free detectors, while avoiding the drawbacks of anchors, rely on extensive post-processing steps to extract the detections (e.g., Non-Maximum Suppression (NMS), thresholding and max-pooling). These post-processing steps also depend on numerous hyperparameters and design decisions that heavily impact the performances of the network [38].

4.2.3 Mask-based Object Detection

In recent developments in computer vision, MaskFormer [65] and Mask2Former [39] have introduced a new meta-architecture designed to predict per-instance or per-class binary masks. Inspired by these pioneering approaches, methods such as MaskRange [66], Mask3D [67], SPFormer [68], and MaskPLS [69] have emerged. These methods concentrate on predicting point masks to enhance instance segmentation within point clouds.

4.3 Occlusion in Point Clouds

Occlusion poses significant challenges for a variety of 3D tasks. For instance, in grasping, partial observation due to occlusion can hinder grasping quality [70], and improving robustness to occlusion (by recovering the complete shape of an object from a partial view) has shown an impressive 45 % improvement over competing architectures. While this kind of task operates on simpler objects and scenes than in autonomous vehicles, this shows that occlusion remains a serious challenge for deep learning methods and that countering it could yield great improvements.

In the context of autonomous vehicles, occlusion is a critical challenge [1], [4]. LiDAR sensors, in a way, provide only 2.5D information, as they return only the first point the beam hits, with no information available behind the objects. This exacerbates the challenges posed by occlusion, thus motivating the development of better techniques to handle occlusion correctly and efficiently. Moreover, in contrast to other domains like geomatics, where complex outdoor scenes are also analyzed, autonomous vehicle tasks involve highly dynamic scenes. Thus, relying solely on multi-temporal data, as is often done in geomatics, is not sufficient, as dynamic objects are constantly moving [71]. Autonomous vehicle tasks sit at a unique intersection where scenes are both highly complex and dynamic, as well as requiring fast inference time for use on the road.

V2V4Real discusses the limitations of single-LiDAR scans, and one solution is to add additional sensors using multiple vehicles (vehicle-to-vehicle (V2V)). However, this is not always feasible. Similarly, techniques based on vehicle-to-infrastructure (V2I) and vehicle-to-everything (V2X) propose sensor fusion, combining on-vehicle sensors with additional sensors on road infrastructure. Again, this may not always be a viable option. As a result, various methods have been developed to better handle occlusion in point clouds from single-scan LiDAR data. We review here some examples of such methods.

BtcDet [1] reconstructs occluded objects by predicting a per-voxel probability of occupancy. The network's objective is to predict high probabilities of occupancy for voxels where a vehicle would be located. Thus, the network is trained to reconstruct the volume occupied by objects, even when they are partially occluded or suffering from signal misses. The complete shape ground truth is obtained by transforming (i.e., completing and mirroring) objects in the input point cloud to estimate their occupied volume. This predicted 3D occupancy map is then used to refine bounding box proposals, making object detection more precise.

Part-A^2 [72], in addition to predicting bounding boxes, also predicts a per-point intra-object part location. For each object, the network is trained to regress the position of each point relative to the vehicle. The location is regressed within the range of $[0, 1]^3$ spanning the volume of the bounding box of the vehicle. The corner $(0, 0, 0)$ is the bottom front right corner of the box, while $(1, 1, 1)$ is the top rear left corner. Thus, for a point on the front bumper of a car, the intra-object location would be predicted as close to the ground and at the front of the vehicle. This approach encourages the network to learn features compatible with the geometry of the vehicles, even when they are occluded. In essence, it enables the network to extract features that help understand what points represent on the vehicle, regardless of occlusion.

SPG [73] generates points to recover missing parts of objects. These points are then combined with the original point cloud to generate a completed point cloud, facilitating object detection. SPG thus proposes a pre-processing technique that completes point clouds before using another object detection architecture, such as PointPillars [21]. The point prediction module employs an encoder-decoder framework to generate new points that complete objects.

Hu *et al.* [74] leverage the hidden information contained in LiDAR point clouds. When a point is measured in 3D space, it provides information not only about the object it hit but also about the unobstructed space between the LiDAR sensor and the impact point. Therefore, when a LiDAR point is obtained, it implies that the space between the LiDAR and the impact point is empty. Hu *et al.* [74] utilize this information to compute a visibility map through raycasting, encoding valuable information about what is occluded or not. This information is then combined with the input point cloud to assist in object detection.

4.4 Datasets

LiDAR datasets play a pivotal role in advancing research and development in the field of autonomous vehicles and robotics. These datasets provide a diverse range of real-world scenarios and challenges that enable researchers to train, validate, and benchmark their deep learning models for tasks like object detection, segmentation, localization, and mapping. We present here some noteworthy LiDAR datasets and summarize them in Table 1, but first let's review what makes autonomous vehicle datasets stand apart from other point cloud datasets.

LiDAR datasets for autonomous vehicles contrast sharply with other types of point cloud datasets. Object datasets are collections of point clouds generated from scanned objects [7], [8]. Being generated from scans, they contain points from all around the objects and are thus way simpler to handle than outdoor scenes encountered in autonomous driving scenarios. Indoor point cloud datasets are also simpler because indoor point clouds often have less variability of object types and denser point clouds [3]. Autonomous vehicle datasets, which are the main interest of this proposition, are typically sparser (as the range is larger) and entail much more complexity.

KITTI The KITTI dataset [75] is a widely recognized benchmark for 3D object detection and tracking in autonomous driving. It includes LiDAR data, stereo images, and annotations for object

instances in urban driving scenarios. The dataset contains 16 124 instances in the training set, and 16 608 in the validation set.

SemanticKITTI Building upon the foundation of KITTI, SemanticKITTI [5] includes full semantic and instance-level annotations for LiDAR scans. This dataset enables semantic segmentation, instance segmentation, and object detection tasks using LiDAR data. SemanticKITTI comprises a total of 22 sequences, with roughly 45 000 scans available

KITTI-360 Liao *et al.* [11] introduced KITTI-360 as a successor to the original KITTI dataset. This evolution incorporates several enhancements, such as additional input modalities and more comprehensive semantic annotations, making it a versatile resource for advancing research in autonomous driving and robotics.

Waymo Open Dataset The Waymo Open Dataset [76] offers a large-scale LiDAR dataset captured from self-driving vehicles in diverse scenarios. Its primary strength lies in its vast scale and variety, making it an invaluable resource for researchers in the field.

nuScenes Comprising 1000 driving scenes captured in urban environments across Boston and Singapore, nuScenes [77] provides a wealth of data for autonomous driving research. It offers semantic, instance, and bounding box annotations to facilitate a wide range of experiments.

Argoverse2 Argoverse2 [12] is a recent development in autonomous vehicle datasets, distinguished by its emphasis on capturing interesting and complex driving scenarios. This dataset also stands out for its diversity, as it gathers data from six cities in the United States and provides detailed bounding box annotations.

V2V4Real V2V4Real [4] presents a novel prospect for V2V. Unlike many V2V datasets that rely on simulations, V2V4Real employs two vehicles, enabling data collection from distinct viewpoints. This unique setup facilitates the capture of occluded regions that might be inaccessible to a single sensor. It's important to note that the annotation in V2V4Real focuses exclusively on vehicles, omitting pedestrians and cyclists.

A9 Intersection Building upon the A9-Dataset [78], A9 Intersection [79] provides valuable data from a unique perspective: the road infrastructure. Specifically, the A9 Intersection collects point clouds from LiDAR affixed to a gantry. Unlike traditional datasets that focus solely on sensor data from vehicles, A9 Intersection explores the possibilities of V2I techniques by relying on sensors embedded in the infrastructure itself. The dataset included calibration between LiDARs and offers 57.4k 3D box labels.

OPV2V OPV2V [80] is a V2V dataset generated from a simulator. This dataset is generated using the CARLA simulator [81], a widely acclaimed platform for simulating autonomous driving scenarios. OPV2V simulates multiple vehicles at a time to capture LiDAR data from multiple viewpoints. The dataset contains approximately 11.5k frames and over 200k vehicle bounding boxes.

V2X-Sim In a similar fashion to the OPV2V dataset, V2X-Sim [82] is another simulation-based dataset. While OPV2V primarily focuses on simulating multiple vehicles within a scene (V2V), V2X-Sim takes a more holistic approach by incorporating not only V2V but also the integration of roadside units, which collect data from an infrastructure perspective (V2I). This amalgamation of both V2V and V2I communication is commonly referred to as V2X. V2X-Sim contains approximately 11k frames.

DAIR-V2X In a significant departure from simulation-based datasets like OPV2V and V2X-Sim, DAIR-V2X, as introduced byYu *et al.* [83], takes a bold leap into the real world to provide a dataset for V2X. What sets DAIR-V2X apart is its incorporation of not just vehicular sensors but also the utilization of infrastructure sensors. This dataset contains approximately 39k frames.

A*3D The A*3D dataset, introduced by Pham *et al.* [84], offers a unique collection of data under challenging conditions. This dataset encompasses a wide range of scenarios, including diverse scenes, varying times of the day, and different weather conditions. Notably, it is designed to include more instances of occlusion, a higher proportion of nighttime frames, and a variety of environmental settings. In total, the A*3D dataset contains seven distinct classes and an impressive collection of approximately 230 000 frames. However, it's worth noting that the dataset is annotated at a relatively low frequency of 0.2 Hz. This sparse annotation frequency can present challenges, particularly for tasks that rely heavily on temporal information. Nonetheless, the dataset's diverse and challenging conditions make it a valuable resource for research and development in the field of 3D object detection and related areas.

Table 1: Comparative table for 3D point cloud datasets. We present here some publicly available datasets, highlighting their key characteristics. We sourced information regarding the number of frames and geographical origins from the research works of Laflamme *et al.* [85] and Xu *et al.* [4]. These datasets fall into three categories: *(i)* Ego datasets when the point clouds are captured from the vehicle's perspective; *(ii)* V2V when multiple vehicles are capturing point clouds concurrently; and *(iii)* V2I in which sensors installed on infrastructure (e.g., gantries) are used for data collection.

Dataset	Year	Source	Type	#Frames (k)	Location
KITTI [75]	2013	Real	Ego	15	Karlsruhe
nuScenes [77]	2019	Real	Ego	40	Boston, Singapore
SemanticKITTI [5]	2019	Real	Ego	43.6	Karlsruhe
Waymo Open Dataset [76]	2019	Real	Ego	200	United States
A*3D [84]	2020	Real	Ego	39	Singapore
KITTI-360 [11]	2022	Real	Ego	80	Karlsruhe
A9-Dataset [78]	2022	Real	Ego	0.5	Munich
OPV2V [80]	2022	Sim	V2V	11	CARLA [81]
V2X-Sim [82]	2022	Sim	V2V & V2I	10	CARLA
DAIR-V2X [83]	2022	Real	V2I	39	Beijing, CN
Argoverse2 [12]	2023	Real	Ego	6000	United States
V2V4Real [4]	2023	Real	V2V	20	Ohio
A9 Intersection [79]	2023	Real	Ego	4.8	Munich

5 Objectives and Methodology

This proposal focuses on investigating the concept of occlusion within 3D LiDAR point clouds for object detection, specifically autonomous vehicle applications. As discussed in Section 2.2, occlusion presents significant challenges for deep learning methods when processing point clouds. There are two main types of occlusion to consider. Firstly, self-occlusion occurs when the near side of an object hides its far side, and this is unavoidable in single-scan LiDAR point clouds. Secondly, external-occlusion happens when obstacles block the laser from reaching the objects, resulting in occluded regions where no points for entire regions of the point cloud. Additionally, incomplete LiDAR scans can occur due to signal misses caused by the laser beam being reflected away from the sensor due to the low reflectance of surfaces. These various types of occlusion and signal miss significantly exacerbate the challenges associated with object detection in point clouds [1]. While occlusion has been extensively studied in objects and indoor datasets with readily available complete ground truth data, there is a significant research gap in addressing these challenges for outdoor scenes due to their complexity and lack of ground truth data. The complexity inherent in outdoor environments, particularly urban landscapes, contrasts sharply with the relative simplicity characterizing indoor spaces like rooms and buildings [3]. This complexity lends itself to unique challenges in applying deep learning techniques to outdoor scenarios, further underscoring the need to address occlusion challenges in such intricate settings.

The main goal of this proposal, which comprises three projects, is to develop novel neural network architectures and innovative techniques to improve LiDAR-based object detection in occluded scenarios. The first project, discussed in Section 5.1, aims to address occlusion within LiDAR point clouds by training a neural network to simultaneously detect and complete objects using BEV instance masks. The second objective, outlined in Section 5.2, is to create a novel multi-LiDAR dataset in an urban context. The novelty of this dataset lies in providing occlusion-free point clouds in complex real-world scenarios, thus offering a unique platform for exploring the influence of occlusion on object detectors. The third project, detailed in Section 5.3, is to develop a novel multi-modal self-supervised learning algorithm using the multi-LiDAR point clouds dataset developed in the second project. Finally, Section 5.4 details the resources required to fulfill the proposed projects.

5.1 Joint Object Detection and Completion

Occlusion poses a significant challenge in deep learning applications on single-scan LiDAR point clouds. It causes objects to appear incomplete in scans, thus impacting the accuracy of object detection algorithms [1]. The first project of this proposal aims to address the issue of occlusion by simultaneously detecting and completing objects in LiDAR point clouds.

One effective approach to handle this challenge is to harness the BEV representation, which naturally suits object detection for autonomous vehicles [18]. Through the formulation of the object detection task into a binary mask prediction framework, the proposed method, referred to as

MaskBEV, predicts a set of BEV instance masks corresponding to the complete footprint of each detected vehicle within a LiDAR scan. We specifically focus on the detection of vehicles due to their larger dimensions, rendering the accurate determination of their actual footprint challenging due to self-occlusion.

Figure 4 illustrates MaskBEV's main ideas. Specifically, Figure 4a shows the mask ground truths overlayed on the BEV input point cloud. Figure 4b shows MaskBEV's output, a set of binary masks representing the completed footprint of vehicles. Each instance is predicted on a separate mask, here shown using different colours. The predicted masks accurately capture the complete footprint of vehicles, even with partial information (i.e., occluded point cloud) about the objects' dimensions and shapes. Figure 4c-h displays the individual raw mask prediction from MaskBEV before the binarization. Each mask prediction only detects a single instance (i.e., the white outlines) and actively suppresses the detection of other instances (i.e., the black outlines). Using this framework, MaskBEV is trained to fight the effects of occlusion by actively predicting the complete footprint of objects from occluded point clouds.

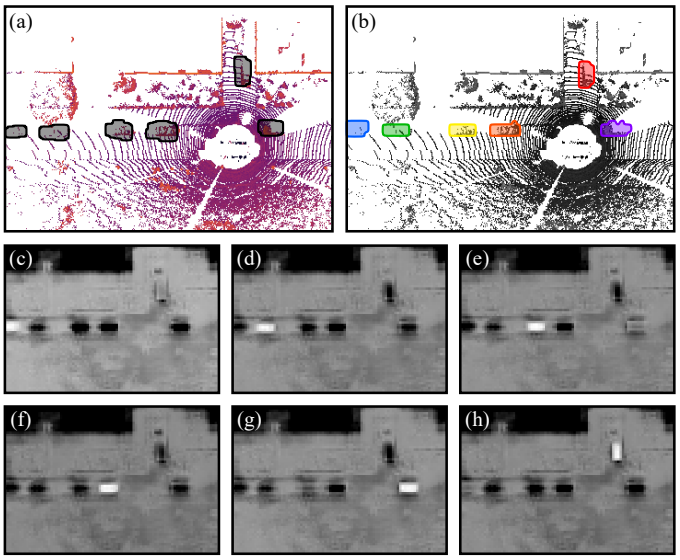


Figure 4: Mask prediction from MaskBEV. (a) Ground truth masks of vehicles' footprints overlayed over a BEV point cloud from the validation set of SemanticKITTI. (b) Mask predictions from MaskBEV. Each vehicle is represented by a distinct mask, distinguished here by different colours. (c-h) Raw mask predictions from MaskBEV, showing the normalized raw scores predicted by the network. Each mask corresponds to a single instance, visible as a white outline. Black outlines represent vehicles that were detected but actively suppressed by the network for a particular mask, indicating that each mask specializes in detecting only one instance and suppresses others. Figure reproduced from [46] © 2023 IEEE.

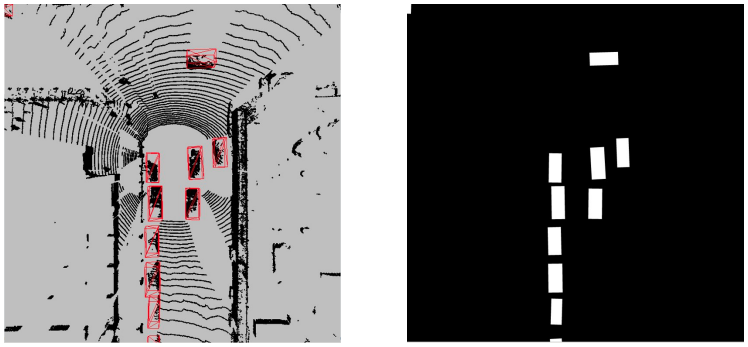
As a testament to its advancement, the first part of this project has been accepted to IROS 2023, the IEEE/RSJ International Conference on Intelligent Robots and Systems, under the title "MaskBEV: Joint Object Detection and Footprint Completion for Bird's-eye View 3D Point Clouds" [46]. We give here an overview of the article and its main contributions.

Firstly, since MaskBEV requires mask ground truth for training, we discuss the mask generation process in Section 5.1.1. Secondly, we give an overview of MaskBEV's architecture in Section 5.1.2. Section 5.1.3 discusses the future works we wish to accomplish on MaskBEV. Finally, the main contributions of this first project are detailed in Section 5.1.4, while the envisioned schedule for their realization is presented in Section 5.1.5.

5.1.1 Ground Truth Masks Generation

MaskBEV requires BEV instance masks for training. From a BEV perspective, an object's footprint is defined as the space it occupies when orthogonally projected onto the ground plane. The ground truth masks are complete, meaning that they accurately represent the full physical footprint of the instance, and not only the footprint of what is visible from a single scan. BEV masks are not commonly available in datasets, to overcome this limitation, we propose an instance mask generation algorithm that generates complete BEV masks from readily available labels. Our mask generation algorithm can produce BEV instance masks from bounding boxes or semantic labels.

For instances annotated with bounding boxes, the process involves extracting 3D bounding boxes, akin to the ones available in datasets such as KITTI [75], and projecting these boxes orthogonally onto the ground plane. This process yields, for each LiDAR scan, a set of binary images on which a single instance — a rectangle — is drawn. The masks generated in this manner lose most of the objects' fine geometry. An example of bounding boxes, and their resulting masks, are shown in Figure 5.



(a) (b)

Figure 5: Mask generation from bounding box labels. (a) Point cloud from Waymo Open Dataset [76] with bounding boxes drawn around cars. (b) Resulting masks generated from the bounding boxes in (a) Each instance corresponds to a distinct mask, shown here in one image for simplicity.

For semantically annotated datasets, such as SemanticKITTI [5], we employ a different approach to generate instance masks. We voxelize the point cloud and use the voxels containing instance points to estimate the objects' footprint, by projecting them onto the ground plane. However, as shown in Figure 6a, the point clouds captured from a LiDAR only capture the LiDAR-facing side of objects. As a result, our algorithm would generate incomplete masks (i.e., masks that do

not represent the full footprint of instances), as illustrated in Figure 6b. Across all instances in SemanticKITTI's validation split, the average ratio between single-scan (i.e.,incomplete masks) and the full footprint is approximately 40 % [46]. This metric underscores that points acquired from a single scan only represent around 40 % of the complete footprint of an instance.

To address this limitation and achieve more accurate masks, we merge multiple sequential LiDAR scans into a map of the environment using their relative pose. This merging process creates a point cloud with points from various viewpoints, allowing us to gather information from all around static vehicles, as depicted in Figure 6c. With this merged point cloud, we can then generate complete instance masks, as shown in Figure 6d. As expected, the resulting mask is now complete and accurately estimates the real footprint of the vehicles. These masks are more precise and can capture object geometries beyond mere rectangles. It is important to emphasize that, despite the use of multiple point clouds for mask generation, the network only receives a single scan as input. Since the masks are generated from the merged point cloud, we exclusively use static objects to train the network. It is worth noting that, although training exclusively features static objects, MaskBEV retains its capacity to detect moving vehicles within the validation set.

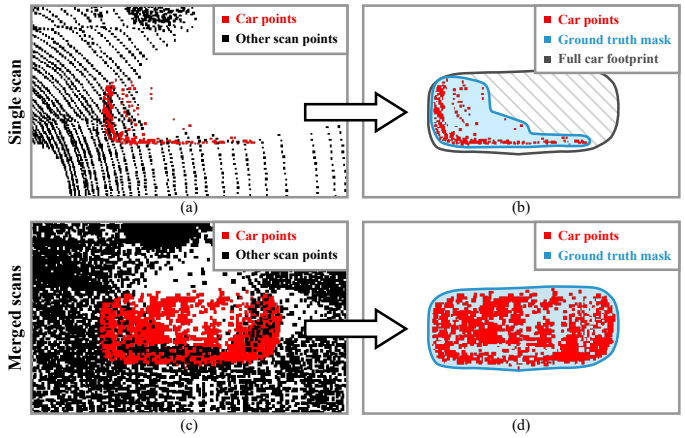


Figure 6: Mask generation from instance label. (a) Point cloud generated from a single LiDAR scan. Red points denote a car, highlighting that solely LiDAR-facing surfaces are visible. (b) Corresponding mask generated from the scan in (a). This mask is partial, failing to encapsulate the full instance footprint. (c) Merged point clouds from successive LiDAR scans. Using multiple scans allows us to gather points from various perspectives, including all sides of vehicles. (d) Resulting mask derived from the merged point cloud in (c). The mask is complete, i.e., it represents the entire footprint of the vehicle. Figure reproduced from [46] © 2023 IEEE.

5.1.2 MaskBEV

As outlined in Section 5.1, MaskBEV offers an innovative approach to object detection in LiDAR point clouds by predicting BEV instance masks using the Mask2Former framework [39]. This approach diverges from the prevalent focus on bounding box predictions in recent object detection

research for LiDAR point clouds presented in Section 4.2. Modern object detection techniques in LiDAR point clouds often revolve around predicting bounding boxes, necessitating substantial prior knowledge about the target objects to achieve accurate results. Both anchor-based and anchor-free detectors demand the use of anchor boxes or intricate post-processing steps, introducing complexities and requiring the tuning of various hyperparameters.

Anchor-based detectors, including single-stage and two-stage variants, rely on numerous anchor boxes to encompass all possible object dimensions and locations. These anchor boxes come with various hyperparameters that need tuning for achieving satisfactory performance, such as dimensions, aspect ratio, and the number of boxes [86].

Anchor-free detectors forgo the use of anchor boxes but instead rely on post-processing steps, such as non-maximum suppression, thresholding, and max-pooling to refine predictions [38]. These post-processing steps necessitate their own set of parameters and design decisions, further requiring fine-tuning.

By contrast, MaskBEV capitalizes on Mask2Former to predict a set of BEV instance masks, MaskBEV circumvents these limitations for object detection in point clouds. MaskBEV's approach bypasses the need for anchors and intricate post-processing, resulting in a more streamlined and efficient object detection framework.

An overview of MaskBEV's architecture is presented in Figure 7. The architecture comprises two core components: an encoder and a mask-prediction module. The encoder transforms 3D point clouds into 2D BEV feature maps, effectively transposing 3D detection into a computer vision task. The encoder sparsely voxelize the input point clouds into pillars, similarly to [21]. Then, each pillar is encoded using a multilayer PointNet [17]. The mask-prediction module then generates a collection of up to M masks, alongside their associated class (i.e., a binary classification of whether there is a car or not in the corresponding mask). For more in-depth information, Guimont-Martin *et al.* [46] provides further details on the MaskBEV architecture.

5.1.3 Analysis of the network

While the initial stages of this project were accepted at the IEEE/RSJ International Conference on Intelligent Robots and Systems, we aim to extend the study of MaskBEV further. Our primary objective is to undertake a comprehensive ablation study of the MaskBEV network and the associated training methodologies. This study aims to elucidate the impact of key components including the encoder, backbone, and mask-prediction modules. Regarding the encoder, we intend to explore various types of encoder networks. Currently, the encoder employs a per voxel multi-layer PointNet [17], and we plan to investigate alternative approaches, such as point convolution, graph-based, and attention-based methods. Additionally, we will subject the detection head to ablation experiments to assess its influence on the overall system performance and compare it with anchor-based and anchor-free approaches. We hypothesize that MaskBEV's performance could significantly improve with larger datasets as seen in computer vision [87], as its transformer-based architecture reduces inductive biases compared to predefined rectangular anchors. We thus wish to extend MaskBEV

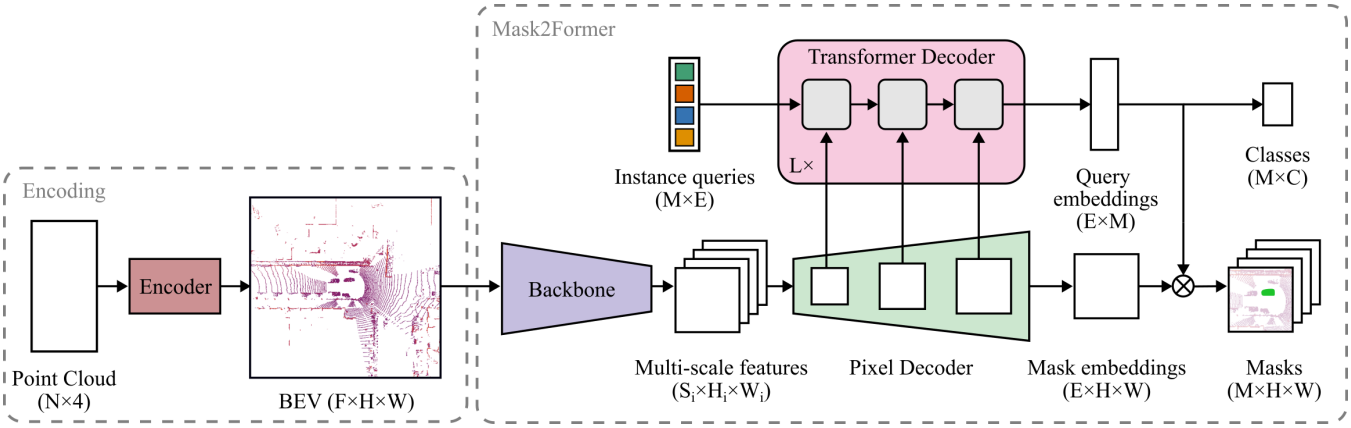


Figure 7: MaskBEV complete architecture. MaskBEV is comprised of two main components: an encoder and a mask-prediction module. The encoder transforms 3D point clouds into BEV feature maps. The mask-predicting module, then predicts a set of classes and BEV masks using Mask2Former [39]. Figure reproduced from [46] © 2023 IEEE.

on larger datasets such as Waymo Open Dataset [76] and nuScenes [77]. An essential aspect of our future investigation is to analyze how the network leverages scene structure since it is useful for completing occluded objects.

Our working hypothesis is that detection is conditioned by both the local features (i.e., the points) and the global structure of the scene (i.e., capturing spatial organization). Given the transformer-based's larger receptive fields [88], we conjecture that these models can more effectively harness global scene structure to enhance detection performance. To validate this hypothesis, we will conduct data augmentation experiments, introducing disruptions to local patterns (e.g., point jitter) or distortions to the global structure (e.g., randomly placing objects within the scene, deliberately out of place). Furthermore, we aim to thoroughly assess the robustness of MaskBEV in handling occlusion. This entails computing the detection metrics with respect to various levels of occlusion and simulating occlusion to the scene using data (i.e., adding occlusion by removing frustums of points in LiDAR scans) to glean insights into system performance under diverse scenarios and conditions.

5.1.4 Objectives

To summarize, the proposed contributions of this first project are:

- a reformulation of object detection in point clouds using mask prediction;
- a novel architecture, MaskBEV;
- an evaluation and ablation study of MaskBEV on various autonomous vehicle datasets; and
- a study of how MaskBEV can leverage both local and global features.

5.1.5 Schedule

The realization of this first project is divided into several tasks described below.

***O1*: Joint Object Detection and Completion**

P1.1 Generation of a BEV mask dataset (completed);

P1.2 Development of the architecture (completed);

P1.3 Benchmark on multiple datasets ***(Conference publication 1)*** (completed); and

P1.4 Ablation study and further analysis ***(Journal publication 1)***

Details of the schedules for these tasks are provided in Section 6 within the timeline of this proposal.

5.2 Multi-LiDAR BEV Dataset

As discussed in Section 4.4, most autonomous vehicle datasets are derived from single-vehicle sources. This often leads to point cloud data that is riddled with occlusions and limited in terms of short-range perception [4]. On the other hand, datasets that do involve multiple LiDAR scanners are typically generated using LiDAR simulators, which often fail to capture the complexity of real-world scenarios both in terms of object types (i.e., the real world is an open set object detection task, which is not possible in simulation) and scenes structure. To address these limitations, recent datasets like V2V4Real [4] used multiple vehicles to capture point clouds from two viewpoints concurrently, allowing each vehicle to capture what might be occluded from the other's perspective. However, V2V4Real has its own set of limitations. Firstly, it confines the relative position of each vehicle to be one directly in front of the other, consequently limiting the amount of data showcasing side-to-side views. Secondly, the dataset only includes annotations for vehicles, omitting more complex objects such as pedestrians and cyclists. In light of these shortcomings, our objective is to create an innovative dataset that addresses these limitations and offers new data collection techniques. The second project in this proposal encompasses two main objectives: the creation of a novel mask dataset derived from V2V4Real, named Masks4Real, and the development of a new dataset capturing point clouds from multiple angles, termed the BEVOcc dataset.

Masks4Real will harness the two viewpoints of V2V4Real to overcome the limitation of the mask generation algorithm of MaskBEV presented in Section 5.1.1. The additional viewpoint will help capture the occluded portions of objects, enabling the generation of masks for dynamic vehicles.

The BEVOcc dataset is designed to address the limitations of existing autonomous vehicle datasets, particularly in terms of occlusions, viewpoints and object types. While V2V4Real predominantly captures data with LiDARs positioned from front to back, essentially one car following the other, our dataset takes a different approach. We focus on a side-by-side arrangement of LiDARs, enabling us to provide complementary perspectives to V2V4Real. This side view is crucial

because it captures objects that may remain occluded in V2V4Real. Moreover, BEVOcc will provide labels for cyclists and pedestrians. This broader annotation scope will enable the generalization of approaches like MaskBEV, discussed in Section 5.1, to different classes of objects and dynamic objects. BEVOcc will be collected using a static LiDARs setup that captures vehicles from both sides of the streets, enabling the generation of point clouds with less occlusion. Additionally, the dataset incorporates an aerial camera viewpoint, embracing the BEV perspective literally. This aerial perspective provides valuable ground truth information regarding objects' footprints, streamlining the annotation process. To further expedite the annotation process, we will leverage state-of-the-art computer vision models such as SAM [89]. These models will generate initial labels, which can then be refined and seamlessly projected onto the point clouds. This ease of annotation is facilitated by the alignment of the BEV viewpoint with the point clouds captured using the static LiDAR setup.

Moreover, this dataset will facilitate investigations into how a smaller, cost-effective, and complementary dataset can enhance object detection performance for point clouds object detection. We plan to strategically gather data to address weaknesses or limitations present in various existing datasets. To mine a diverse range of interesting scenarios, the BEVOcc dataset will be collected in busy streets, intersections, and city centers. This approach aligns with previous work, such as [12], which underscores the benefits of focusing on intriguing scenarios in autonomous vehicle datasets.

Furthermore, the BEVOcc dataset aims to explore multi-modality by incorporating additional sensors. This could involve polarized cameras, super-resolution imaging, hyperspectral cameras, and other sensing modalities. This will enable future work on co-training and self-supervised learning to harness the potential benefits of combining data from multiple sources during training.

To summarize, the first goal is to derive a new mask dataset from V2V4Real, described in Section 5.2.1. This dataset aims to enhance the performance of mask-based object detection, especially for moving vehicles. The second goal is to complement the existing data landscape with a new multi-angular LiDAR dataset, referred to as the BEVOcc dataset, which will be detailed in Section 5.2.2. The data collection for the BEVOcc dataset involves careful sensor calibration, which will be discussed in Section 5.2.4. Furthermore, the acquisition pipeline will be detailed in Section 5.2.3. The development of an annotation pipeline to provide precise annotations for various objects in the dataset will be described in Section 5.2.5. Finally, the effectiveness of the BEVOcc dataset will be evaluated by benchmarking against current state-of-the-art models and integrating the new data, as discussed in Section 5.2.7.

5.2.1 Masks4Real: a mask dataset using V2V4Real

The first step of this project involves the creation of a new BEV masks dataset named Masks4Real, which is derived from the V2V4Real dataset. In MaskBEV, due to the need to use sequential LiDAR scans to gather points around from all sides of objects, we were limited to only training on static objects. However, since V2V4Real uses two LiDARs, resulting in two distinct viewpoints of objects in the scene, this allows for the generation of masks, even for dynamic objects.

The Masks4Real dataset addresses the primary limitation of MaskBEV's training by allowing the application of the mask generation algorithm presented in Section 5.1.1 to moving vehicles. Since V2V4Real provides bounding box annotations for various vehicle types (cars, vans, pickup trucks, semi-trucks, and buses), the Masks4Real dataset will allow the generation of masks through two main methods: bounding box projection and point projection.

For the masks generated from bounding boxes, we will project the annotations onto the ground plane. We will utilize data from both LiDAR scans of the vehicles, effectively doubling the available mask labels from V2V4Real. For masks generated from points, we need to adapt the mask generation algorithm from MaskBEV to account for the dual viewpoint data from V2V4Real. This process will involve detecting the ground plane (which can be achieved by excluding points below a certain height threshold), extracting points within each bounding box from both scans and applying MaskBEV's algorithm to generate masks.

The resulting Masks4Real dataset will serve as training data for MaskBEV, providing training samples for moving objects. The inclusion of both bounding box-generated and point-generated masks in this dataset will allow a comparative analysis of each method's performance. Additionally, the Masks4Real dataset will serve as the foundation to which BEVOcc's data will be added, contributing to the enhancement of mask-based object detection in point clouds.

5.2.2 Bird's-Eye View Occlusion (BEVOcc)

The BEVOcc dataset aims to leverage the use of multiple LiDARs to capture street scenes from multiple viewpoints, thereby providing point clouds with minimal occlusion. To obtain a complete view of the scene, an aerial perspective will be captured using a BEV camera, taking the bird's-eye view (BEV) perspective literally. This camera can be mounted on a drone, positioned at a high vantage point, or affixed to a tall tripod. This would generate a view BEV similar to Krajewski *et al.* [90]. For the sake of brevity and clarity, we will henceforth refer to this BEV camera as using a drone in the context of this proposal. This aerial viewpoint will facilitate the annotation process by providing ground truth information about the exact positions and dimensions of objects, which can then be projected back into the point clouds. By leveraging the static nature of our setup, this approach ensures accurate annotation and simplifies greatly the annotation process. In addition to LiDAR sensors, the BEVOcc dataset will incorporate other sensors to explore multi-modal learning. For each LiDAR, there will be a corresponding camera that captures visual information about the scenes. A polarized camera will also be integrated into the setup, which offers valuable information about surfaces, including their physical properties, orientation, reflection angle, and degree of polarization [91]. Furthermore, the BEVOcc dataset will include hyperspectral cameras to capture unique spectral signatures of objects, facilitating advanced material and object recognition [92].

The BEVOcc dataset will contribute to addressing limitations present in existing datasets and enable the exploration of multi-modal learning techniques, as outlined in our third objective discussed in Section 5.3.

Figure 8 shows the dataset collection setup using two acquisition units U_1 and U_2. This arrangement will allow the collection of points from both sides of vehicles, thus providing ground truth information about the complete object's geometry. In addition to LiDARs, other sensors, described in Section 5.2.3, are used to capture additional modalities. The data collection will be done in various weather conditions and scene types, including low and high-traffic scenarios, pedestrian areas, and complex urban environments, with a focus on capturing busy streets with intricate occlusion scenarios involving cars, cyclists and pedestrians.

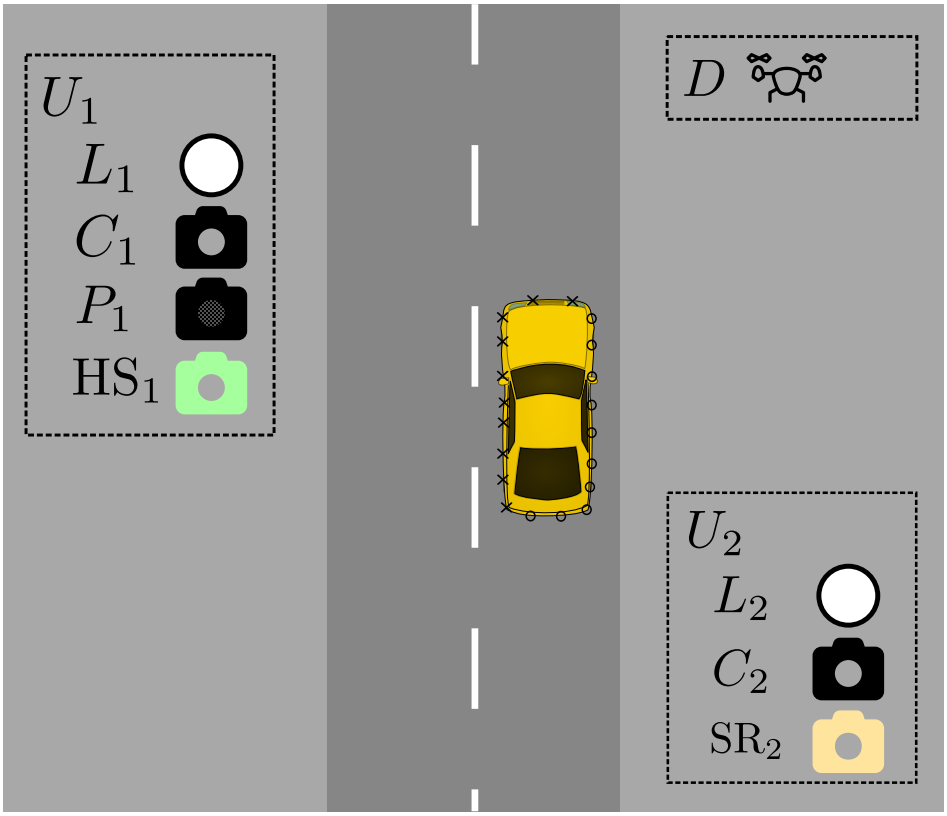


Figure 8: Example of BEVOcc data collection setup. Multiple acquisition units, denoted as U_i, are placed on different sides of streets. This setup allows for the capture of points from multiple viewpoints around objects, illustrated by crosses and circles. Alongside LiDARs, denoted as L_i, cameras (C_i) capture visual information about the scene. Additional sensors, including polarized cameras (P_i), hyper-spectral cameras (HS_i), and super-resolution imaging devices (SR_i), are incorporated into each acquisition unit. A camera denoted as D captures an aerial view of the scene from a BEV perspective.

5.2.3 Acquisition

The acquisition pipeline is illustrated in Figure 9. The pipeline will use the Robot Operating System (ROS) framework for communication and data recording using the `rosbag` tools. The system

consists of multiple acquisition units connected to a central base computer for control. Each acquisition unit is equipped with LiDAR, cameras, and various additional sensors, including polarized cameras (Blackfly S USB3 from Teledyne FLIR), super-resolution cameras, and hyperspectral cameras. These sensors are rigidly mounted together, as the sensor stick used in Helmberger *et al.* [93], to simplify the inter-sensor calibration process described in Section 5.2.4.

Each acquisition unit establishes a wireless connection to a base station, which is controlled by the central base computer. To avoid wireless throughput issues, each unit records its data locally onto its storage device (typically an SSD). Lower-frequency data will be transmitted to the base station for monitoring the data acquisition process.

The base station consists of a computer, a Wi-Fi access point for sending commands to acquisition units, and a GPS-based Network Time Protocol (NTP) server to ensure accurate timing information. The drone employed in the setup is the DJI Mini 2 [1], controlled using a smartphone. This model, manufactured by DJI, is equipped with features that make it well-suited for data acquisition tasks. Notably, the drone features built-in stabilization through a gimbal, enhancing the stability of data collection activities. It is capable of recording high-quality video, offering options such as 4K (3840×2160) video recording at 30 Hz or 2.7K (2720×1530) video recording at 60 Hz.

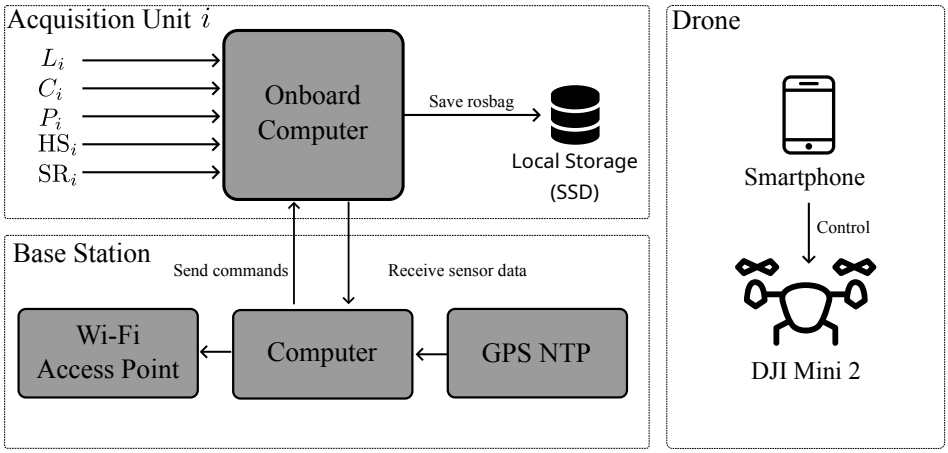


Figure 9: Data flow of the proposed acquisition system. The base station includes a laptop computer, a Wi-Fi access point for connecting to acquisition units, and a GPS NTP server for precise timing. An example acquisition unit U_i is shown, gathering data from all its sensors (LiDAR L_i, camera C_i, polarized camera P_i, hyperspectral camera HS_i, and super-resolution camera SR_i) using an onboard Jetson Nano computer. The acquisition unit records data in `rosbag` format onto a local SSD. The drone, a DJI Mini 2, is controlled using a smartphone.

The design of the acquisition units will be created using computer-aided design (CAD) software. These units will be designed to be easily expandable for adding additional sensors as required, such as solid-state lidars or event cameras. CAD files of the acquisition units will be made publicly available to facilitate wider adoption and future modifications.

[1] dji.com/ca/mini-2

5.2.4 Calibration

The calibration process for the multi-sensor acquisition system involves several steps to ensure accurate and consistent data alignment. This section outlines the calibration procedures for both intrinsic and extrinsic calibrations, as well as the calibration between different acquisition units.

Acquisition Unit Calibration Each sensor within an acquisition unit requires intrinsic calibration to correct for its internal parameters. Additionally, the sensors need to be extrinsically calibrated with respect to the LiDAR, which is the referential of each unit. For cameras, a chessboard pattern will be used for calibration. To achieve extrinsic calibration between the LiDAR and camera sensors within each unit, a target-based calibration method [94] will be employed. This will yield the homogeneous transformation matrix between the LiDAR's origin and the camera's viewpoint. Given the rigid attachment of sensors within each unit, the camera-LiDAR calibration can be performed off-field before deployment. Existing tools such as CalibrationTools [2] from TIER IV, Inc., SensorsCalibration [3] from Shanghai AI Laboratory, or the ROS package cam_lidar_calibration [95] [4] can be used for the camera-LiDAR calibration process.

Time Synchronization Each acquisition unit will be synchronized with the base station using the NTP protocol, employing the `chrony` [5] tool. To ensure synchronization between the drone's image data and the base station, GPS time will be used, as obtained from a GPS NTP Network Time Server on the base station.

Extrinsic Calibration Between Units Once each acquisition unit is individually calibrated, the next step involves determining the extrinsic calibration between the units while gathering data. We start by picking one acquisition unit, U_1, as a reference point from which all other positions will be relative. The first step is to align the LiDAR's ground plane of U_1 with the z = 0 plane using CalibrationTools. Drawing inspiration from [96], LB-L2L-Calib [97] will be employed for the relative calibration of LiDAR sensors. This technique uses a calibration sphere that is moved within the field of view of all LiDARs. During the acquisition process, we will move into the field of view of both acquisition units with the sphere to allow the calibration of the relative position between the units. Alternatively, the iterative closest point (ICP) matching algorithm from libpointmatcher [98] can be employed for matching static point clouds captured by different LiDARs. In both cases, this calibration process yields homogeneous transformation matrices representing the pose of each LiDAR relative to a reference LiDAR, U_1.

[2] CalibrationTools
[3] github.com/PJLab-ADG/SensorsCalibration
[4] github.com/acfr/cam_lidar_calibration
[5] chrony-project.org

Drone Calibration The drone's camera will also be calibrated using a chessboard pattern. The relative position of the aerial drone with respect to U_1 will be computed using large visual fiducials, such as AprilTags [99], placed on top of the acquisition units. These fiducials will allow accurate localization of the drone relative to the acquisition units. Figure 10 shows an example of a drone picture taken at an altitude of 50 m, which is necessary to capture large streets such as Boulevard Laurier near Laval University's campus. The AprilTag, with an approximate side length of 0.5 m, on the picnic table in the picture can be detected, and thus used for localization. The tag used for the acquisition units will be larger, and in most cases seen from a lower altitude (i.e., around 25 m for most streets), thus allowing for more precise localization.

Figure 10: Example of an aerial image taken from a drone. Picture taken from a DJI Mini 2 drone flying at an altitude of 50 m above the Grand Axe on Laval University's campus. The bicycle lanes visible on both sides of the images are approximately 50 m apart. The AprilTag on the picnic table is detectable from this picture and will be made bigger for the data collection process.

All extrinsic calibration processes described above will yield homogeneous transformation matrices. Given two referential U_1 and U_2, ${}^{U_1}_{U_2}H$ the pose of U_2 in U_1's referential, and an homogeneous point ${}^{U_2}P = \begin{pmatrix} x & y & z & 1 \end{pmatrix}$ in U_2's referential, it is possible to find ${}^{U_1}P$, the position of the same point in U_1's referential.

The extrinsic calibration matrix between U_1 and U_2 will have the following form

$$ {}^{U_1}_{U_2}H = \begin{pmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \\ 0_{1\times3} & 1 \end{pmatrix}, $$

where $\mathbf{R}_{3\times3}$ is the rotation matrix between the referential and $\mathbf{T}_{3\times1}$ is the position of U_2 in U_1's referential. We can then transform the point ${}^{U_2}P$ into U_1's referential using ${}^{U_1}P = {}^{U_1}_{U_2}H {}^{U_2}P$. This will allow us to transform all points in U_1's referential to simplify the annotation process.

5.2.5 Annotation

The annotation process for the proposed dataset is outlined in Figure 11. The key idea is to utilize the aerial camera data from the drone to detect objects of interest and then project these detections onto the ground plane in the point cloud. This approach enables the generation of annotations such as bounding boxes, BEV masks, or per-point semantic and instance labels at a relatively low annotation cost. The annotations can be refined using an algorithm similar to Krajewski *et al.* [90] that uses consecutive frames to reduce false positive detections in BEV highway images.

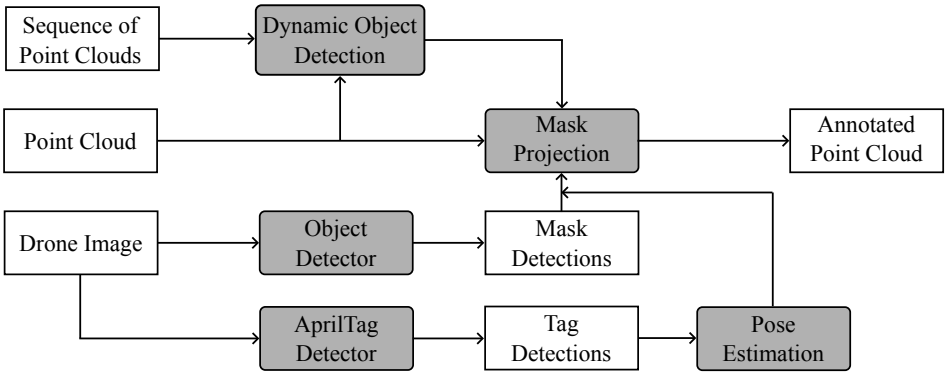


Figure 11: Annotation process of the proposed dataset. Object detections from the drone's image are projected onto the point cloud using the drone's pose determined from AprilTags on each acquisition unit. The projections are further refined using dynamic objects detected within the entire acquisition sequence.

The annotation pipeline involves several steps to achieve accurate annotations within the point cloud. We describe here the outline of each step.

1. Find the ground plane in U_1's referential using the LiDAR point cloud.
2. Detect AprilTags in the aerial image captured by the drone.
3. Estimate the drone's pose relative to acquisition unit U_1 using the detected AprilTags.
4. Detect objects in the drone's image using a state-of-the-art computer vision object detector.
5. Project the object detections from the image onto the point cloud using the drone's pose information and intrinsic calibration, as done in Côté *et al.* [100].
6. Refine the projected annotations using dynamic object information found throughout the acquisition sequence.

7. Extract points above the projected annotation and derive semantic, bounding box and mask annotations.

8. Manually validate the annotations to remove any errors using tools like SUSTechPOINTS [6]

The annotation process is further clarified with pseudocode in Algorithm 1. Figure 12 illustrates the projection of image labels onto the point cloud, detailing the steps for projecting annotations from the drone's image onto the ground plane of the point cloud.

Algorithm 1 Pseudocode of the annotation process

Input: P the point cloud to be annotated, the drone's image I, a sequence S of LiDAR scans, the intrinsic calibration of the drone's camera C
Output: Annotation for P
$G \leftarrow$ Detect ground plane in P using the sequence S
$D_{obj} \leftarrow$ Find dynamic objects in P using the sequence S
$T \leftarrow$ Detect AprilTags in I
$D \leftarrow$ Estimate the pose of the drone using T
$O \leftarrow$ Find objects in I using a state-of-the-art object detector
$M \leftarrow$ Empty array of size length(O)
$i \leftarrow 1$
for all $o \in O$ **do**
  $m \leftarrow$ Project o onto the ground plane G using C and D
  $m' \leftarrow$ Refine projection using D_{obj}
  $p \leftarrow$ Extract points above the ground plan and within m'
  $m'' \leftarrow$ Derive bounding box, semantic and mask annotations from p
  $M[i] \leftarrow m''$
  $i \leftarrow i + 1$
end for
return M

5.2.6 Development kit

To facilitate the wider adoption of the BEVOcc dataset within the research community, we intend to release a dedicated Python development kit. This kit aims to simplify the process of accessing and effectively utilizing the dataset, regardless of the specific deep learning framework chosen by researchers. The development kit will be made available on the Python Package Index (PyPI) [7] for easy installation and integration. By providing a consistent and unified interface for accessing and processing the BEVOcc dataset, the development kit aims to lower the entry barrier for researchers while enhancing the dataset's usability and impact in the field.

[6] github.com/naurril/SUSTechPOINTS
[7] pypi.org/

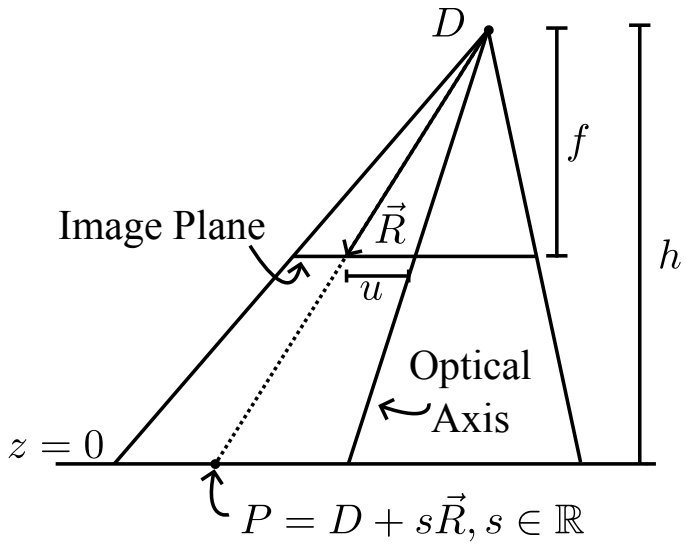


Figure 12: Projection of image labels onto the point cloud. We show here a simplified 2D version of the projection, which is trivially generalizable to 3D. Given the drone's position D and its intrinsic calibration (focal length f of the camera), we project an annotation from position u in the image (relative to the camera's optical axis) onto the $z = 0$ plane. Let $\vec{R}$ be the vector starting at D and pointing to the annotation's position: $\vec{R} = \begin{pmatrix} u & -f \end{pmatrix}^\mathsf{T}$. We then find the intersection between the line $D + s\vec{R}$ ($s \in \mathbb{R}$) and the ground plane $z = 0$. For mask detection, this process can be applied to the contour points of the mask to derive the complete projected mask.

5.2.7 Benchmark

The Masks4Real and BEVOcc datasets aim to address several research questions through comprehensive benchmarking and evaluation of state-of-the-art models and MaskBEV. The first of which is to investigate the impact of incorporating additional orthogonal data, obtained from a different sensor setup and at a lower annotation cost, to enhance the performance of existing object detection models. Specifically, we seek to understand whether supplementing a larger dataset with focused, smaller-scale BEVOcc data can help improve object detection results. Furthermore, the BEVOcc dataset allows to generalize MaskBEV to dynamic objects, pedestrians and cyclists, allowing further study of the mask-based object detection framework in point clouds. This benchmarking will provide valuable insights into the dataset's potential to enhance object detection across different object categories, aiding in the development of more comprehensive and accurate detection models in occluded scenarios.

5.2.8 Objectives

To summarize, the proposed contribution of this second project are:

- development Mask4Real, a new mask-based dataset derived from V2V4Real;
- development BEVOcc, a novel multi-point of view LiDAR and multi-modal dataset; and
- benchmark of state-of-the-art networks and MaskBEV.

5.2.9 Schedule

The realization of this second project is divided into several tasks described below.

***O2*: Multi-angular point cloud dataset for outdoor object detection**

P2.1 Development of Mask4Real;

P2.2 Development of the acquisition platform;

P2.3 Data collection;

P2.4 Data annotation;

P2.5 Development the development kit; and

P2.6 Benchmark networks using the dataset ***(Publication 2)***

Details of the schedules for these tasks are provided in Section 6 within the timeline of this proposal.

5.3 Multi-Modal Self-Supervised Learning

Humans can readily comprehend occlusion, capitalizing on their inherent understanding of 3D scene structure. Self-Supervised Learning (SSL) is believed to imbue neural networks with a form of such rudimentary background knowledge or common sense [13]. Hence, our objective is to employ SSL to instill in the network a proficient prior regarding the 3D scene structure within an autonomous vehicle context.

Self-Supervised Learning (SSL) has emerged as an invaluable tool to circumvent the laborious and expensive process of data labelling [3]. SSL allows neural networks to learn feature representation without relying on any human-generated labels, and instead learn directly from the data without relying on any human-generated labels. As data collection is usually cheaper than its annotation, it becomes very interesting to pre-train on a large quantity of unlabelled data to acquire robust features, and subsequently fine-tune on a smaller, labelled, dataset to accomplish a specific task. This paradigm is particularly interesting for point cloud datasets, where annotation costs can be prohibitive due to the complexity of annotating 3D data [5], [11]. Notably, these techniques offer the potential to extend the application of point cloud-based object detection models to environments and regions for which no large annotated datasets are available. With many autonomous vehicle datasets primarily collected in major urban centers of North America and Europe, this could hinder the application of deep learning methods to other regions or environment types. Reducing the annotation burden with SSL, and the data collection cost as done with BEVOcc, could allow the broader application of object detectors in contexts such as rural environments, developing countries, or forests.

The third project in this proposal is to develop a novel SSL training technique based on the Joint-Embedding Predictive Architectures (JEPA) framework [101], [102], aimed at making neural networks better at understanding occlusion. BEVOcc, developed in the second project, will serve as a solid foundation to experiment with the JEPA framework to train object detectors in LiDAR point clouds. The crux of this project is to align the features between LiDAR scans and aerial drone-captured BEV images. The BEV perspective provides invaluable information about the scene's actual structure that is not present in LiDAR point clouds. Our working hypothesis is that using SSL technique to extract features coherent between both representations of the same physical scene will yield better performances for object detection in complex and occluded scenes, even if the BEV is not used during inference.

In the annotation process described in Figure 11, only a fraction of the information available in the drone's aerial image is used for training — the output of the computer vision object detector — before being discarded and not used for training. Using SSL techniques would allow the extraction of more information from the data, and could thus enhance detection performances. Furthermore, auxiliary sensors like ground-level cameras, hyperspectral cameras, and polarized cameras remain untapped resources, offering the prospect of a multi-modal self-supervised framework to improve point cloud object detector performances. This novel SSL technique could be used with the BEVOcc dataset, and with the other available datasets.

The goal of this third project is to harness these modalities within a self-supervised framework to enhance performance. We first desire to exploit the BEV perspective available in BEVOcc by applying techniques like JEPA [102] due to the contrasting nature of LiDAR point clouds and aerial imagery. Subsequently, we aim to expand this framework to encompass a wider array of sensor types.

5.3.1 Joint-Embedding Predictive Architectures

Our first objective is to apply the JEPA framework [101], [102] to BEV images and point clouds. The core principle of JEPA departs from conventional SSL methods based on joint-embedding (i.e., learn to predict similar embedding for compatible inputs), such as DINOv2 [103], or the reconstruction paradigm (i.e., try to reconstruct the input from a corrupted view), like MAE [104] or PointBERT [41]. JEPA's distinctive approach is to apply a generative method directly in embedding space, and not the input space (e.g., pixels, points).

We give here an overview of the JEPA framework [102]. Given two variables, x and y, which could be two different modalities capturing the same scene (e.g., a point cloud and a BEV image), two encoders are employed to yield representations s_x and s_y. The architecture then learns to predict s_y (the representation of y) from s_x (the representation of x). The prediction is potentially conditioned by a latent variable z. The latent variable z serves as a means to introduce information for predicting s_y that might not be present in s_x. For example, z could encode temporal information that is not present in a single LiDAR scan. The energy function used for learning is $E(x, y, z) = D(s_y, Pred(s_x, z))$, which we wish to minimize for compatible inputs. Unlike conventional reconstruction-based approaches, the emphasis here is not on replicating every nuance of y; rather, it focuses on capturing relevant information. Additionally, the incorporation of VICReg can prevent the collapse of representations (i.e., the energy function simply outputting a constant value for all inputs) [105], further bolstering the framework's effectiveness.

JEPA stands out as a self-supervised learning technique capable of seamlessly operating across distinct modalities [102]. This framework affords the possibility of performing SSL from images to point clouds, facilitating knowledge transfer between different data representations. Moreover, the integration of a temporal dimension is conceivable, with the ability to track individual instances across multiple frames.

5.3.2 JEPA on BEVOcc

The proposed integration of the Joint-Embedding Predictive Architectures (JEPA) framework with the **B**ird's-**E**ye **V**iew **Occ**lusion (BEVOcc) dataset is illustrated in Figure 13. The core concept revolves around leveraging both the LiDAR point cloud and the drone's bird's-eye view (BEV) image within the JEPA framework. Since both data modalities capture the same underlying scene, it is reasonable to anticipate the ability to predict one modality's representation from the other. The goal is to train the point cloud encoder to extract embeddings that align with the BEV perspective

of the scene, providing occlusion-free and comprehensive ground truth information about the scene's global structure. Additionally, this framework could also leverage the temporal relation between point clouds, by tracking instances in time and aligning their features across multiple sequential scans.

To apply this self-supervised technique to BEVOcc, we will employ the JEPA framework to corresponding pairs of BEV voxels and image patches. Consequently, the network will be trained to predict the representation of the aerial image's features from the corresponding voxel in the point cloud. In other words, segments of the point cloud that correspond to the same spatial region as an image patch will serve as the input for the JEPA process, as illustrated in Figure 14. We plan on using Transformer-based networks to accurately leverage global information about the scene.

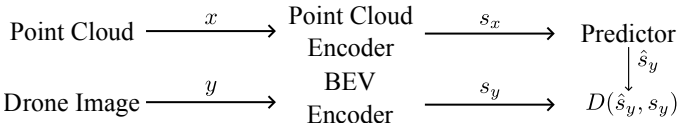


Figure 13: Proposed application of the JEPA framework to the BEVOcc dataset. A point cloud encoder will extract features s_x from an input point cloud. A BEV encoder will extract features s_y from an input aerial image. We then predict s_y from s_x, meaning that we train the network to learn features s_x compatible with the BEV image of the same scene.

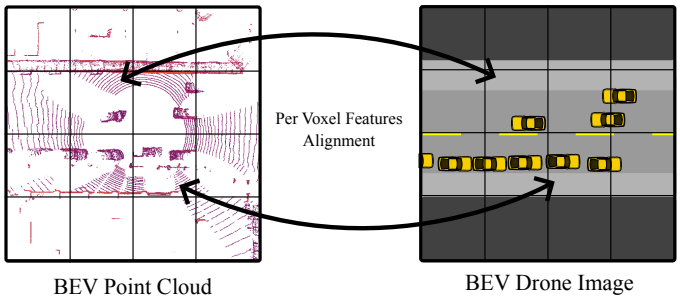


Figure 14: Integration of the JEPA framework with the BEVOcc dataset. The point cloud encoder generates feature embeddings (s_x) from the point cloud input. Simultaneously, the BEV encoder produces feature embeddings (s_y) from the aerial image input. The network is trained to predict s_y from s_x, effectively learning features within the point cloud that are compatible with the BEV perspective of the same scene.

5.3.3 Multi-modal JEPA

In addition to the application of JEPA to the drone's image and point cloud, we also aim to extend JEPA into a multi-modal framework. The main idea of this approach is illustrated in Figure 15.

In addition to predicting the embedding of the drone's image, we intend to predict the embeddings of other sensor modalities, such as the cameras attached to each acquisition unit. This concept could be further expanded to include additional modalities like the hyperspectral and polarized cameras present within BEVOcc. Even if these sensors are not present at inference time on an actual autonomous vehicle, this technique could be used as a form of Learning using Privileged Information (LUPI), where extra knowledge is used during training [106]. We refer to this kind of training as *sensory scaffolding*, as additional modalities are used as "scaffolding" to "build" good feature representation.

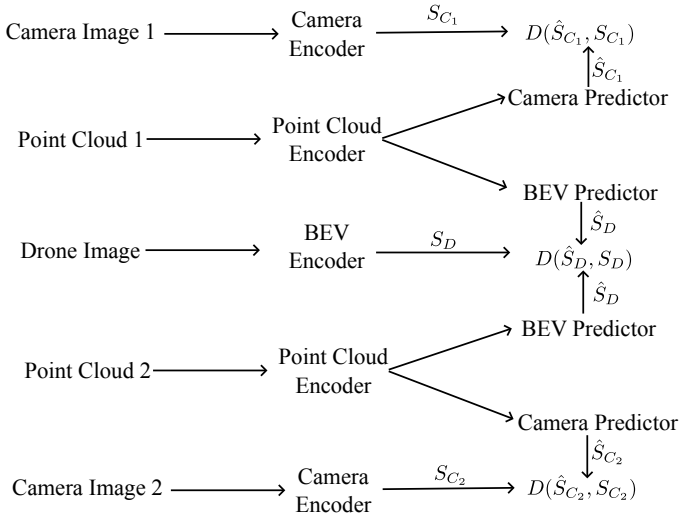


Figure 15: Extension of the JEPA framework to a multi-modal context. The proposed approach involves the application of JEPA across multiple modalities found within the BEVOcc dataset. As elaborated in Section 5.3.1, we seek to predict not only the embedding of the drone's image but also those of other sensors, such as the cameras associated with each acquisition unit. S_D refers to the feature representation of the drone image, S_{P_i} to the features of the ith point cloud, and S_{C_i} to those of the ith camera. The $\hat{S}_{\circledcirc}$ refers to the prediction of $S_{\circledcirc}$ from another modality, and $D(\cdot, \cdot)$ is the energy function to be minimized.

5.3.4 Objectives

To summarize, the proposed contribution of this third project are:

- Creation of a new SSL training method based on JEPA;
- Generalization of this technique to a multi-modal context; and
- Evaluation of the technique on BEVOcc.

5.3.5 Schedule

The realization of this third project is divided into several tasks described below.

***O3*: Multi-modal self-supervised learning**

P3.1 Development of the SSL technique on drone images and point cloud data;

P3.2 Adaptation of the SSL technique to other modalities; and

P3.3 Evaluate the network performances ***Publication 3***

Details of the schedules for these tasks are provided in Section 6 within the timeline of this proposal.

5.4 Resources

In order to accomplish the objectives outlined in Section 5, several critical resources will be required, both in terms of computational capabilities and equipment.

Given the substantial computational demands associated with training deep neural networks, this project proposal requires access to a range of high-performance NVIDIA GPUs, each equipped with a minimum of 42GB of memory. The Norlab at Laval University will provide a workstation equipped with an NVIDIA Quadro RTX 8000 GPU, primarily intended for development purposes. For intensive training tasks, a compute server equipped with four NVIDIA RTX A6000 GPUs will be accessible for model training and experimentation, allowing a high throughput of training operations. This server will serve as a powerful platform for model training and experimentation. To complement our GPU requirements, resources from Calcul Québec [8] and Compute Canada [9] will be used.

The data collection of BEVOcc, as described in Section 5.2, will require specific sensors. The Norlab already possesses a large inventory of sensors, including LiDARs and polarized cameras. This inventory will allow us to quickly start the data collection process.

[8] calculquebec.ca
[9] alliancecan.ca

6 Schedule

Figure 16 presents the schedule for each objective outlined in Section 5. Each objective is accompanied by the main tasks required to accomplish them. The first objective, presented in Section 5.1, was addressed during the autumn 2022 and winter 2023 semesters, leading to a conference publication [46]. Subsequently, work on the second objective, presented in Section 5.2, will commence in the autumn semester of 2023. The third objective, presented in Section 5.3, will begin once data from the second objective starts to be collected. This approach allows for agile data collection and enables adaptation based on the needs of the research.

Although the order of these objectives might seem confusing at first glance, the development of the first objective, MaskBEV, served as a foundation for the subsequent projects. It raised several research questions about occlusion and its handling in state-of-the-art methods, which ultimately inspired the creation of a new dataset, the second objective. The data collection for the second objective will be conducted in an agile manner to detect possible issues or areas for improvement early in the development process. Finally, this new dataset will serve as the perfect medium to develop novel self-supervised techniques, described in the third objective.

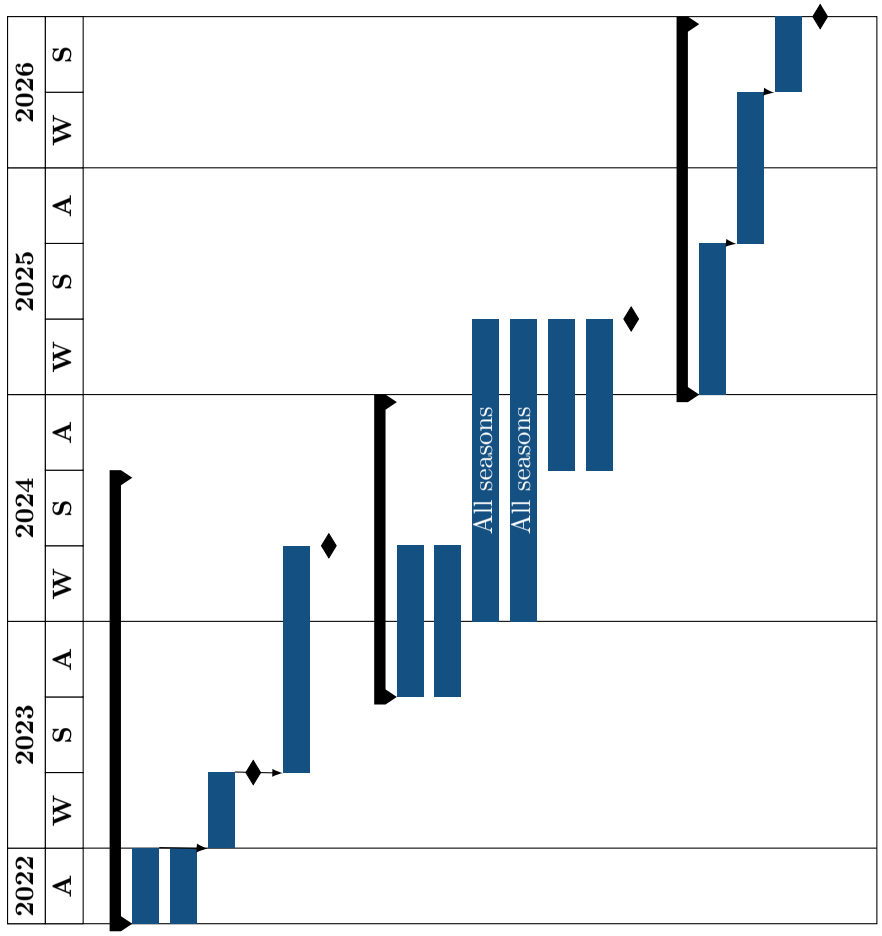


Figure 16: Proposed schedule for achieving the proposed objectives and tasks described in Section 5. The numbers refer to the objectives listed in Section 5. W= winter semester, S = summer semester, and A: Autumn semester.

7 Conclusion

This proposal aims to enhance our understanding of the concept of occlusion in LiDAR point cloud data for deep learning methods. As discussed in Section 2 and Section 3, occlusion plays a crucial role in point clouds. With the evolution of mobile systems in more complex environments, effectively handling occlusion becomes critical for deep learning methods and might be the key to the future of deep learning on LiDAR point clouds. The objectives presented in Section 5 each take a step towards a better understanding of occlusions in the context of LiDAR point clouds for autonomous mobile systems. The first objective reformulates the object detection task as a mask prediction and completion problem. The second objective aims to provide a novel and versatile BEV dataset to enable a comprehensive study of occlusion and open new research possibilities. Lastly, the third objective builds on the previous two to propose a new self-supervised technique that leverages the BEV nature of the developed dataset. Finally, these objectives are planned in a three-year schedule in Section 6.

References

[1] Q. Xu, Y. Zhong, and U. Neumann, "Behind the Curtain: Learning Occluded Shapes for 3D Object Detection," in *AAAI Conference on Artificial Intelligence*, 2022.

[2] S. A. Bello, S. Yu, C. Wang, J. M. Adam, and J. Li, "Deep learning on 3D point clouds," *Remote Sensing*, vol. 12, no. 11, p. 1729, 2020.

[3] B. Fei, W. Yang, L. Liu, *et al.*, "Self-supervised Learning for Pre-Training 3D Point Clouds: A Survey," *arXiv preprint arXiv:2305.04691*, 2023.

[4] R. Xu, X. Xia, J. Li, *et al.*, "V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 712–13 722.

[5] J. Behley, M. Garbade, A. Milioto, *et al.*, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9297–9307.

[6] J.-L. Déziel, P. Merriaux, F. Tremblay, *et al.*, "Pixset: An opportunity for 3d computer vision to go beyond point clouds with a full-waveform lidar dataset," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 2987–2993.

[7] Z. Wu, S. Song, A. Khosla, *et al.*, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[8] A. X. Chang, T. Funkhouser, L. Guibas, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[9] I. Armeni, O. Sener, A. R. Zamir, *et al.*, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1534–1543.

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

[11] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2022.

[12] B. Wilson, W. Qi, T. Agarwal, *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," *arXiv preprint arXiv:2301.00493*, 2023.

[13] Y. LeCun and I. Misra, *Self-supervised learning: The dark matter of intelligence*, Accessed Aug, 2023. https://ai.meta.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/, 2021.

[14] W. Guimont-Martin, "Apprentissage par réseaux de neurones profonds sur les nuages de points 3d," Accessed Aug, 2023. https://willguimont.github.io/assets/papers/ApprentissageParReseauxDeNeuronesProfondsSurLesNuagesDePoints3D.pdf.

[15] Z. Wu, S. Song, A. Khosla, *et al.*, "3D ShapeNets: A Deep Representation for Volumetric Shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

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

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

[18] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660.

[19] C. Choy, J. Gwak, and S. Savarese, "4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

[20] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[21] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[22] L. Zhang, J. Sun, and Q. Zheng, "3D Point Cloud Recognition Based on a Multi-View Convolutional Neural Network," *Sensors*, vol. 18, no. 11, p. 3681, 2018.

[23] H. You, Y. Feng, R. Ji, and Y. Gao, "Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1310–1318.

[24] G. Pang and U. Neumann, "3D point cloud object detection with multi-view convolutional neural network," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, 2016, pp. 585–590.

[25] H. Su, V. Jampani, D. Sun, *et al.*, "Splatnet: Sparse lattice networks for point cloud processing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2530–2539.

[26] Y. Rao, J. Lu, and J. Zhou, "Spherical fractal convolutional neural networks for point cloud recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 452–460.

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

[28] A. Boulch, G. Puy, and R. Marlet, "FKAConv: Feature-kernel alignment for point cloud convolution," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[29] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and Deformable Convolution for Point Clouds," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6411–6420.

[30] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep Convolutional Networks on 3D Point Clouds," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019, pp. 9621–9630.

[31] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on X-Transformed Points," *Advances in neural information processing systems*, vol. 31, 2018.

[32] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[33] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining Point Cloud Local Structures by Kernel Correlation and Graph Pooling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4548–4557.

[34] C. Chen, G. Li, R. Xu, T. Chen, M. Wang, and L. Lin, "ClusterNet: Deep Hierarchical Cluster Network with Rigorously Rotation-Invariant Representation for Point Cloud Analysis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4994–5002.

[35] Y. Zhang and M. Rabbat, "A Graph-CNN for 3D Point Cloud Classification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 6279–6283.

[36] G. Te, W. Hu, A. Zheng, and Z. Guo, "RGCNN: Regularized Graph CNN for Point Cloud Segmentation," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 746–754.

[37] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[38] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, Springer, 2020, pp. 213–229.

[39] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 864–17 875, 2021.

[40] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.

[41] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 313–19 322.

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[43] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2906–2917.

[44] T. Guan, J. Wang, S. Lan, *et al.*, "M3detr: Multi-representation, multi-scale, mutual-relation 3d object detection with transformers," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2022, pp. 772–782.

[45] Z. Liu, H. Tang, A. Amini, *et al.*, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 2774–2781.

[46] W. Guimont-Martin, J.-M. Fortin, F. Pomerleau, and P. Giguère, *MaskBEV: Joint Object Detection and Footprint Completion for Bird's-eye View 3D Point Clouds*, 2023.

[47] J. Mao, Y. Xue, M. Niu, *et al.*, "Voxel Transformer for 3D Object Detection," in *CVPR*, 2021.

[48] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast Encoders for Object Detection from Point Clouds," in *CVPR*, 2019.

[49] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-Based 3D Single Stage Object Detector," in *CVPR*, 2020.

[50] W. Zheng, W. Tang, L. Jiang, and C.-W. Fu, "SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud," in *CVPR*, 2021.

[51] Y. Zhang, Q. Zhang, Z. Zhu, J. Hou, and Y. Yuan, "Glenet: Boosting 3d object detectors with generative label uncertainty estimation," *arXiv preprint arXiv:2207.02466*, 2022.

[52] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *NIPS*, 2015.

[53] Y. Chen, S. Liu, X. Shen, and J. Jia, "Fast Point R-CNN," in *CVPR*, 2019.

[54] S. Shi, L. Jiang, J. Deng, *et al.*, "PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection," *International Journal of Computer Vision*, pp. 1–21, 2022.

[55] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection," in *AAAI Conference on Artificial Intelligence*, 2021.

[56] Q. Xu, Y. Zhong, and U. Neumann, "Behind the Curtain: Learning Occluded Shapes for 3D Object Detection," in *AAAI Conference on Artificial Intelligence*, 2022.

[57] G. Wang, J. Wu, B. Tian, S. Teng, L. Chen, and D. Cao, "CenterNet3D: An Anchor Free Object Detector for Point Cloud," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 953–12 965, 2021.

[58] Y. Hu, Z. Ding, R. Ge, *et al.*, "AFDetV2: Rethinking the Necessity of the Second Stage for Object Detection from Point Clouds," in *AAAI Conference on Artificial Intelligence*, 2022.

[59] Q. Chen, L. Sun, Z. Wang, K. Jia, and A. Yuille, "Object as Hotspots: An Anchor-Free 3D Object Detection Approach via Firing of Hotspots," in *ECCV*, 2020.

[60] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-Based 3D Object Detection and Tracking," in *CVPR*, 2021.

[61] P. Sun, M. Tan, W. Wang, *et al.*, "SWFormer: Sparse Window Transformer for 3D Object Detection in Point Clouds," in *ECCV*, 2022.

[62] J. Li, H. Dai, L. Shao, and Y. Ding, "Anchor-free 3D Single Stage Detector with Mask-Guided Attention for Point Cloud," in *ACM-MM*, 2021.

[63] R. Ma, C. Chen, B. Yang, *et al.*, "CG-SSD: Corner Guided Single Stage 3D Object Detection from LiDAR Point Cloud," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 191, pp. 33–48, 2022.

[64] T. Zou, G. Chen, Z. Li, *et al.*, "KAM-Net: Keypoint-Aware and Keypoint-Matching Network for Vehicle Detection From 2-D Point Cloud," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 207–217, 2021.

[65] B. Cheng, A. Schwing, and A. Kirillov, "Per-Pixel Classification is Not All You Need for Semantic Segmentation," *NeurIPS*, 2021.

[66] Y. Gu, Y. Huang, C. Xu, and H. Kong, "MaskRange: A Mask-classification Model for Range-view based LiDAR Segmentation," *arXiv preprint arXiv:2206.12073*, 2022.

[67] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, "Mask3D for 3D Semantic Instance Segmentation," *arXiv preprint arXiv:2210.03105*, 2022.

[68] J. Sun, C. Qing, J. Tan, and X. Xu, "Superpoint Transformer for 3D Scene Instance Segmentation," *arXiv preprint arXiv:2211.15766*, 2022.

[69] R. Marcuzzi, L. Nunes, L. Wiesmann, J. Behley, and C. Stachniss, "Mask-Based Panoptic LiDAR Segmentation for Autonomous Driving," *IEEE Robotics and Automation Letters*, 2023.

[70] B. Sen, A. Agarwal, G. Singh, B Brojeshwar, S. Sridhar, and M. Krishna, "SCARP: 3D Shape Completion in ARbitrary Poses for Improved Grasping," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 3838–3845.

[71] J. Barros-Ribademar, J. Balado, P. Arias, and S. M. González-Collazo, "Visibility analysis for the occlusion detection andcharacterisation in street point clouds acquired withMobile Laser Scanning," *Geocarto International*, vol. 37, no. 25, pp. 10 152–10 169, 2022.

[72] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From Points to Parts: 3D Object Detection From Point Cloud With Part-Aware and Part-Aggregation Network," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2647–2664, 2020.

[73] Q. Xu, Y. Zhou, W. Wang, C. R. Qi, and D. Anguelov, "SPG: Unsupervised Domain Adaptation for 3D Object Detection Via Semantic Point Generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 446–15 456.

[74] P. Hu, J. Ziglar, D. Held, and D. Ramanan, "What You See is What You Get: Exploiting Visibility for 3D Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 001–11 009.

[75] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[76] P. Sun, H. Kretzschmar, X. Dotiwalla, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[77] H. Caesar, V. Bankiti, A. H. Lang, *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[78] C. Creß, W. Zimmer, L. Strand, *et al.*, "A9-Dataset: Multi-Sensor Infrastructure-Based Dataset for Mobility Research," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2022, pp. 965–970.

[79] W. Zimmer, C. Creß, H. T. Nguyen, and A. C. Knoll, "A9 Intersection Dataset: All You Need for Urban 3D Camera-LiDAR Roadside Perception," *arXiv preprint arXiv:2306.09266*, 2023.

[80] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 2583–2589.

[81] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Conference on robot learning*, PMLR, 2017, pp. 1–16.

[82] Y. Li, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2X-Sim: A Virtual Collaborative Perception Dataset for Autonomous Driving," *arXiv preprint arXiv:2202.08449*, 2022.

[83] H. Yu, Y. Luo, M. Shu, *et al.*, "DAIR-V2X: A Large-Scale Dataset for Vehicle-Infrastructure Cooperative 3D Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 361–21 370.

[84] Q.-H. Pham, P. Sevestre, R. S. Pahwa, *et al.*, "A 3d dataset: Towards autonomous driving in challenging environments," in *2020 IEEE International conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 2267–2273.

[85] C.-É. N. Laflamme, F. Pomerleau, and P. Giguere, "Driving datasets literature review," *arXiv preprint arXiv:1910.11968*, 2019.

[86] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 734–750.

[87] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 104–12 113.

[88] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 116–12 128, 2021.

[89] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[90] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems," in *2018 21st international conference on intelligent transportation systems (ITSC)*, IEEE, 2018, pp. 2118–2125.

[91] W. Fan, S. Ainouz, F. Meriaudeau, and A. Bensrhair, "Polarization-based car detection," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, pp. 3069–3073.

[92] M. Jürise, A. Udal, J. Kaugerand, and R. Sell, "Hyperspectral camera with polarized filter as modern supersensor device for cyber-physical systems," in *2018 16th Biennial Baltic Electronics Conference (BEC)*, IEEE, 2018, pp. 1–4.

[93] M. Helmberger, K. Morin, B. Berner, N. Kumar, G. Cioffi, and D. Scaramuzza, "The hilti slam challenge dataset," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7518–7525, 2022.

[94] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 5562–5569.

[95] D. Tsai, S. Worrall, M. Shan, A. Lohr, and E. Nebot, "Optimising the selection of samples for robust lidar camera calibration," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 2631–2638.

[96] Z. Qiu, J. Martínez-Sánchez, P. Arias-Sánchez, and R. Rashdi, "External multi-modal imaging sensor calibration for sensor fusion: A review," *Information Fusion*, p. 101 806, 2023.

[97] J. Zhang, Q. Lyu, G. Peng, Z. Wu, Q. Yan, and D. Wang, "LB-L2L-Calib: Accurate and robust extrinsic calibration for multiple 3D LiDARs with long baseline and large viewpoint difference," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 926–932.

[98] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP Variants on Real-World Data Sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, Feb. 2013.

[99] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *2011 IEEE international conference on robotics and automation*, IEEE, 2011, pp. 3400–3407.

[100] S. Côté and W. Guimont-Martin, *Aerial cable detection and 3D modeling from images*, US Patent 11,521,357, 2022.

[101] M. Assran, Q. Duval, I. Misra, *et al.*, "Self-supervised learning from images with a joint-embedding predictive architecture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 619–15 629.

[102] Y. LeCun, "A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27," *Open Review*, vol. 62, 2022.

[103] M. Oquab, T. Darcet, T. Moutakanni, *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[104] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[105] A. Bardes, J. Ponce, and Y. LeCun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," *arXiv preprint arXiv:2105.04906*, 2021.

[106] J. Lambert, O. Sener, and S. Savarese, "Deep learning under privileged information using heteroscedastic dropout," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8886–8895.